Formalisation and Evaluation of Focus Theories for Requirements Eliciation Dialogues in Natural Language

Renaud Lecœuche

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy to the University of Edinburgh
1999

Declaration

This thesis has been composed by myself. It has not been submitted in any previous application for a degreee. The work reported within was executed by myself unless otherwise stated.

Renaud Lecœuche
February 10, 2000

Acknowledgements

I would like to thank my thesis supervisors, Dr. Catherine Barry from the Institut National des Sciences Appliquées in Rouen, Mr. Dave Robertson and Dr. Chris Mellish, from the University of Edinburgh. They provided insightful and constructive comments on often crude ideas. I really appreciated their availability to discuss my work, and their regular enquiries on the progress made.

I would also like to thank my family for their support.

The Software Systems and Processes Group provided an agreeable and stimulating environment. I would like to thank its members for the regular discussions and seminars we have had.

Finally, I would like to thank the Edinburgh Eagles Handball Club for the many hours of training and the games we played during the last three years.

Contents

List of Algorithms

List of Figures

List of Tables

Notation

In the rest of this thesis, we distinguish some elements in the main text with a special notation. The notation is summarised in the following table.

Notation used in the thesis

Notation	Interpretation
#identifier	Identifier of an element in the specification The entity or relation referred to by the identifier
“system output”	Output produced by the system
`input`	Input entered by the user
`specification`	A part of the specification
`communication`	A communication
?hypothetical	An hypothetical element of the specification

List of Acronyms

We also use some acronyms. They are presented below:

DRS .... Discourse Representation Structure

EG .... Eloquence Generation

ER .... Entity-Relationship

FIFO .... First-In-First-Out

IRU .... Information Redundant Units

WWW........ World-Wide-Web

Chapter 1

Introduction

Objectives

The objectives of this chapter are to:

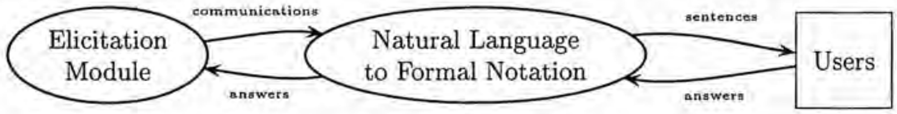
- Present the contribution of this thesis,
- Present related work in requirements engineering,
- Present the organisation of the thesis.

1.1 Contribution

Requirements engineering is an important part of software engineering. It consists in defining the needs of users when building a new system. These needs may be functional, i.e., what service should the system be able to provide, as well as non-functional, i.e., under which constraints should the system operate. Errors in requirements may have disastrous effects in the rest of the software engineering process (Brooks 1995, p.199), since they would lead to the construction of a system of little interest to its users or would require expensive modifications to correct. Because requirements documents may be very large, errors are usually hard to detect manually. Computer support is therefore often beneficial for their analysis. This is made easier if requirements are expressed formally. However, this support must also be adapted to and be usable by people who are expressing their requirements. These people are usually not computer specialists and are not accustomed to use formal languages. It is therefore necessary to help them express their requirements. Numerous approaches, have been suggested as aids to the acquisition of requirements (Reubenstein 1990). Much less attention has been paid to the control of the dialogue taking place between the users and the system whilst using such frameworks (Bubenko et al. 1994). Frameworks for requirements acquisition are not normally accompanied by theories of the types of dialogue which they support. Our ability to develop sophisticated formal frameworks to analyse requirements makes this deficiency more acutely felt, since increases in formality are often accompanied by greater difficulty in understanding and using the frameworks (Robertson et al. 1989).

The architecture of most existing requirements elicitation systems can be depicted as in figure 1.1 (the meanings of "communications", "sentences" and "answers" are discussed

Figure 1.1 Existing general architecture



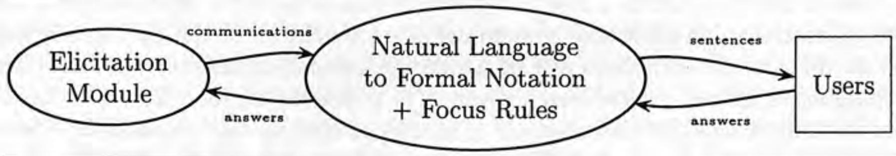
later in the thesis). Users write their requirements in more or less natural language. This is then translated into a formal language that can be interpreted by the elicitation module. This module works on the requirements and provide feedback. The translation process is then applied to convert feedback into more or less natural language. Different systems put different emphasis on the parts of that general architecture. Some are very good at natural language interpretation while others put more emphasis on analysing the requirements and providing feedback.

Natural language approaches to requirements elicitation, put an emphasis on natural language interpretation (see section 1.2.1). In these approaches, users write their specification in a subset of natural language. The system then translates it into a formal notation. The main benefit provided by these approaches is the improvement in the ease of use of the system: natural language is the main means of communication for human beings and does not need to be learned. However, most of these approaches do not provide a dialogue well suited for the requirements elicitation process. Because they translate the natural language specification into a formal notation but do not provide guidance on how to write the specification in the first place, users are left in charge of writing correct requirements. If a mistake is made while writing the specification, it will simply be translated into the formal notation.

In order to actively help users in the process of writing the requirements, the elicitation system must interact with them. The emphasis, here, is no longer on translating requirements, but on actively extracting them through a dialogue with users. This is useful, since the requirements elicitation process is complex, and offering guidance is a big help for users. Unfortunately, most of the approaches providing guidance expose their formal underlying frameworks directly to users (see section 1.2.2). In order to benefit from the guidance provided, users have to learn the idiosyncrasies of the system they use. The task of providing guidance is complicated by the fact that there are numerous ways of carrying out the requirements elicitation. Very little research has been done on how to organise best the elicitation process to provide effective guidance. An arbitrary choice could be made, but forcing users to adopt a predefined method is usually not possible as it would make the elicitation process very difficult to follow and understand. The system must therefore be able to adapt itself to various elicitation methods. On the other hand, it is necessary for the system to make choices in order to provide active guidance. A "least-commitment" strategy, such as asking users at every choice point what to do next, is not a useful approach (Ferguson et al. 1996).

One way of offering guidance without restricting users too much is by communicating with them in natural language, and by using natural language constraints to inform the choices made by the system to select a guidance strategy. These constraints ensure that the system adopts a strategy that will guide users in a natural and understandable manner, by taking into account the current state of the dialogue. In other words, the system takes

Figure 1.2 General architecture



Example 1.1 Requirements elicitation dialogue

S1:	Research group #rg1 is the research group.
S2:	Do you want a site presenting it?
U1:	Yes.
S3:	A site presents the research group.
S4:	Which site presents the research group?
U2:	A new site.
S5:	Site #s1 presents the research group.
S6:	Do you want a home page describing site #s1?
U3:	Site #s1 is described by a new home page.
S7:	Home page #hp1 describes site #s1.
S8:	What is the title of home page #hp1?
U4:	Research group.
S9:	Do you want a page linked from it?
U5:	Yes.
S10:	A page is linked from research group home page #hp1.
S11:	Which page is linked from research group home page #hp1?
U6:	A new page.
S12:	Page #pa1 is linked from research group home page #hp1.
S13:	What is the title of page #pa1?
U7:	I would like to speak about the publication set
S14:	Publication set #ps1 is the publication set.
S15:	It is involved in the research group.
S16:	Do you want a site presenting it?

into account the current state of the specification to help users complete it, but the current state of the dialogue is the principal factor constraining what will be spoken about next. Using such an approach reduces some of the problems discussed above. The specification does not need to be immediately correct as it will be checked and reworked by the system. The formal framework is hidden from users but is still there to ensure the correctness of the specifications. Guidance is continuously offered through dialogue, which is influenced by but does not directly follow the steps of construction of the specification.

The natural language constraints we use in this thesis are theories of dialogue coherence, called "focus" theories. They define what can be spoken about next in a dialogue based on what has already been discussed and the subject under discussion. The theories take into account what participants in a dialogue pay attention to and try to ensure that the rest of the dialogue is related to it. The general architecture of our system is then as presented in figure 1.2. Focus theories seem well-suited for organising the kind of dialogues we consider here. Take for example the dialogue of example 1.1 between an elicitation system (S) and a user (U) about a research group World-Wide-Web (WWW) site. This dialogue was generated by an elicitation system using focus rules (how this dialogue is generated and the meaning of the identifiers will be explained in detail in chapter 7). The systems tries to help its users define how a research group WWW site should look like. The way the dialogue evolves from discussing the research group, to discussing the site and its associated home page, to discussing the set of publication can quite easily be followed. The use of pronouns helps in making the text fell natural. It would have been difficult to achieve the same result without using focus rules.

Other techniques for organising dialogues, such as those based on the intentions underlying the dialogue (Cohen et al. 1990), would require the dialogue manager to know what the elicitation system is trying to achieve and what its plan is. For some elicitation systems, this knowledge may not be available. Similarly, techniques based on the content of the communications exchanged and how they relate, e.g., based on RST (Mann and Thompson 1987), usually require a lot of domain knowledge. They are therefore time-consumming to code. Focus theories require less information from the elicitation module while enabling the dialogue manager to structure the dialogue. However, in some cases, focus theories are not sufficient to organise a dialogue. We use a theory based on speech act (see section 3.4.1) and some ideas from Grice's work on conversation (see section 5.2.1) to deal with these cases. More generally, although we tried to minimise the impact of other theories to study in detail focus theories, it would be interesting to know whether and how we can integrate them with the work presented in this thesis. In particular, the notion of dialog act and its application to dialog grammar could be of interest. General frameworks developped to study various aspects of dialogue, including dialog acts and focus, have started to appear but work is still at an early stage (C-Star Consortium 1998; Allen and Core 1997).

Organising a dialogue based on attention requires a lot of domain knowledge in order to know how things mentioned in the dialogue relate to each other. Therefore, the amount of knowledge engineering needed to build natural language applications is also an important issue. We have tried to limit the engineering difficulties by clearly separating the domain knowledge needed by our dialogue manager from its management capabilities, and by providing a way of re-using the existing domain knowledge as far as possible. This is done by using rules which enable us to re-use part of the domain knowledge already used by the elicitation module.

The contribution of this thesis is therefore *the formalisation and evaluation of focus theories for requirements elicitation dialogues in natural language.* The main questions we deal with are the following:

- Which focus theories should we use?
- What are the relations between the constraints imposed by the focus theories and the constraints inherent to the requirements elicitation process?
- Does this approach improve the perceived quality of the dialogue between the elicitation tool and its users?

A prototype system has been developed. This system mainly operates in the WWW site design domain. It has also been applied in other domains as an initial demonstration of the range of problems that can be tackled by our approach.

1.2 Related work

Numerous approaches have been concerned with allowing users to input their requirements in natural language and translating them into a formal language. This is important since few users are accustomed to deal directly with a formal notation. Numerous approaches have also been concerned with providing guidance to users during the requirements elicitation process. These approaches have mostly been concerned with ensuring the correctness of the specification, which is a difficult task. Surprisingly, very few approaches have tackled the two issues together.

We give here an overview of an approach concerned with natural language interpretation and an overview of an approach concerned with guidance. These two examples are representative of the large number of other methods in the same fields. We also give an overview of a method concerned with natural language and guidance. This approach is the only one, to our knowledge, to have tackled extensively both aspects at the same time.

1.2.1 Natural language

In this section, we give an overview of ACE (Fuchs and Schwitter 1995, 1996; Fuchs et al. 1998; Schwitter and Fuchs 1996) as an example of an approach translating natural language statements into statements expressed in a formal notation. This process is complex because natural language is ambiguous (Meyer 1985; Macias and Pulman 1993) and informal (Balzer et al. 1978). ACE is a general approach translating natural language statements into Discourse Relation Structure (Kamp 1981; Gamut 1991). Other methods exist for translating natural language into Discourse Relation Structure (Covington 1988), object-oriented frameworks (Mich 1996; Nanduri and Rugaber 1996; Moreno 1997), algebraic specification frameworks (Ishihara et al. 1993), Entity-Relationship (ER) diagrams (Chen 1983), and KAOS (Leite and Franco 1993; Leite 1993) (see section 1.2.2).

ACE is based on a unification-based grammar interpreting a subset of English. The grammar allows constructs such as simple declarative sentences, relative clauses, comparative constructions, if...then sentences, negation, yes/no queries and wh-queries. ACE also deals with simple anaphora resolution. These constructs are then translated into Discourse Relation Structure. Discourse Representation Structure (DRS) is a logical framework well suited for natural language interpretation. Once a natural language specification has been translated into DRS, it can be transformed into Prolog. It can then be executed. This provides important feedback for users and helps them correct the specification if need be (Fuchs 1992).

A problem with this approach is that the system does not provide help on how to write the specification. No error-checking is made. If a mistake is made while writing the specification, it will simply be translated into the formal notation.

1.2.2 Guidance

In this section, we give an overview of KAOS (Dardenne et al. 1993; Darimont 1995; Darimont and van Lamsweerde 1996) as an example of an approach providing guidance to users during the requirements elicitation process. KAOS is a method primarily concerned with functional requirements, i.e., requirements about the services the system under development will have to provide. Other methods exist for other aspects such as non-functional requirements (Chung et al. 1994; Mylopoulos et al. 1992), requirements business rationale (Yu 1994), and scenario (Helm and Fickas 1992; Hsia et al. 1994).

KAOS is based on a model containing what can be spoken about when eliciting requirements. This model contains among others the following notions (the descriptions are taken from Dardenne et al. (1993)):

Entities are autonomous objects whose instances may exist independently from other object instances.

Relationships are subordinate objects whose instances' existence depends upon the existence of the corresponding instances linked by the relationship.

Events are instantaneous objects.

Actions are mathematical relations over objects.

Agents are objects which are processors for some actions.

Goals are non-operational objectives to be achieved by the system.

Constraints are operational objectives to be achieved by the system. As opposed to goals, a constraint is formulated in terms of objects and actions available to some agent in the system.

The main aim of the approach is to fill in this model with a particular problem specification. To achieve this, a method is provided which begins by eliciting goals. These goals are then decomposed and operationalised into constraints. At the same time, other objects such as entities, relationships, events, actions, and agents, are elicited. A major issue during this process is to ensure that the decompositions are correct, i.e., that the resulting constraints ensure that the goals are achieved, and to guide users in choosing a decomposition. In order to do this, goals are expressed in a temporal logic notation and are then classified depending on their pattern (see Thayse (1989, ch. 4) for a description of temporal logic). Generic decompositions are then provided for each class of goals. These decompositions have been proved mathematically correct. This enables users to choose a decomposition well-suited to the kind of goals they are working on and to be sure that the decomposition is mathematically sound. For example, an "achieve" goal ($P \Rightarrow \Diamond Q$) can be reached if we first reach a milestone R ($P \Rightarrow \Diamond R$) which allows us to reach the goal ($R \Rightarrow \Diamond Q$). Using these decompositions, users are able to refine goals until they can be expressed by constraints on agents' actions. This approach therefore provides guidance in the requirements elicitation process by directing the decomposition of goals based on their temporal logic pattern.

A problem with this approach is the need to know about temporal logic in order to benefit from the guidance offered. The framework underlying the specification is directly exposed to users. This may represent a sizeable obstacle as many users may not have enough knowledge about temporal logic to use the method. Moreover, users are still left in charge of deciding what the next goal to work on should be without receiving help. Each single step in the decomposition of goals is supported by the tool but the overall process is not. This could lead to a disorganised interaction between the tool and its users.

1.2.3 Guidance and natural language

As was seen in the two preceding sections, numerous approaches deal with the issues of providing guidance or interpreting natural language during requirements elicitation. However, very few approaches tackle these two issues at the same time. Such an approach was developed in the OICSI project (Rolland and Proix 1992) and then modified in the CREWS project (Ben Achour 1998; Rolland and Ben Achour 1998). In this last approach, natural language is used to specify scenarios. The scenarios can then be used to derive goals that need to be achieved. Each goal is then associated with new scenarios and the process recurs.

The natural language interpretation in this approach is based on a case grammar (Fillmore 1968). The surface structure of a sentence is transformed by the grammar into a semantic pattern. Semantic patterns for simple clauses have the form N(V)[C] where N is the semantics of verb V, such as communication or action, and C is the list of cases associated with it, such as agent, object, source and destination. This pattern is then transformed and added to the specification. This process, though different from the one used in ACE, has the same function.

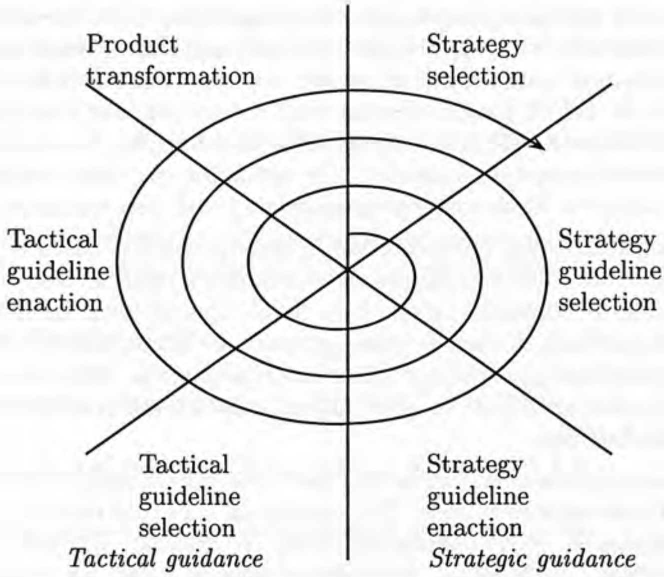
Guidance is provided on two levels. First, the system helps users correct any deficiencies in the way the scenario is written. Deficiencies can appear at two levels: style deficiencies and content deficiencies. Style deficiencies cover, for example, the use of anaphora which may make the scenario ambiguous. Content deficiencies cover, for example, missing elements in the semantic pattern. For example, the sentence `The user inputs the code` corresponding to the semantic pattern `communication(input)[Agent: 'the user'; Object: 'the code'; Source: 'the user'; Destination: ?]` is incomplete. It should be rewritten as `The user inputs the code in the ATM` to complete the `destination` slot. Second the system provides guidance by pointing out possible points where the scenario could be extended. For example, if the scenario contains a test, both the successful and unsuccessful branches of the test should be specified.

More generally, guidance is provided at a tactical level and a strategic level (Rolland 1994; Si-Saïd and Rolland 1998). At the tactical level, the system helps users in carrying out actions to achieve the results they want. Scenario correction is an example of tactical action. At the strategic level, the system helps users in deciding what should be done next. Scenario completion is an example of strategic action. This is shown in figure 1.3.

The strategic guidance consists of choosing a strategy (strategy selection), choosing guidelines to help implement it (strategy guideline selection), and enacting the guidelines by selecting the next step to perform (strategy guideline enaction). The tactical guidance then consists of choosing guidelines to help implement the step (tactical guideline selection), enacting the guidelines by proposing transformations (tactical guideline enaction), carrying out the transformations (product transformation). Once a step has been performed, the strategy selection is carried out again and the cycle recurs. By combining natural language processing and guidance, this approach makes the writing of scenarios potentially easier.

Compared to these approaches, our work puts more emphasis on the dialogue aspect of the interaction between users and system. Although our system checks the specification for correctness, it is less developed than KAOS. Its natural language understanding capabilities are also less developed than those of ACE. On the other hand, our system is able to interact with users. Compared to CREWS, the difference is that guidance is mainly constrained by the evolution of the dialogue. This should make the dialogue more natural to users.

Figure 1.3 Guidance in CREWS (adapted from Si-Saïd and Rolland (1998))



1.3 Organisation

The thesis is organised in six main parts in addition to the introduction:

Review describes the main focus theories proposed by the natural language processing community. Examples of the use of these theories are also presented. This part consists of chapter 2.

Formalisation presents the formalisations of the focus theories we use in our system and how these formal theories intervene in the functionning of the system. The link between the framework used to formalise the theories and the framework used in the system for manipulating requirements is carefully studied. This part consits of chapters 3 and 4.

Implementation is a detailed discussion of the generation and interpretation of natural language in our system, with a special emphasis on the role of focus theories in these processes. This part consists of chapters 5 and 6.

Examples are given in the next part. A worked out example is first presented. It is followed by additional examples showing the use of our system in different domains. This part consists of chapters 7 and 8.

Evaluation gives a precise evaluation of the approach proposed in the thesis. It also deals with the development method we used to build our system. It consits of chapters 9 and 10.

Conclusion concludes the thesis and discusses issues left for further research. It consists of chapter 11.

Table 1.1 Thesis organisation

Introduction	Introduction (ch. 1)	
Review	Focus Theories (ch. 2)	
Formalisation	Formal Theories (ch. 3)	System Architecture (ch. 4)
Implementation	Generating Outputs (ch. 5)	Interpreting Inputs (ch. 6)
Examples	Worked Example (ch. 7)	Additional Examples (ch. 8)
Evaluation	Evaluation Results (ch. 9)	Development Method (ch. 10)
Conclusion	Conclusion (ch. 11)	

This is summarised in table 1.1. All chapters are introduced by a description of their objectives.

Chapter 2

Focus Theories

Objectives

The objectives of this chapter are to:

- Define the notion of focus,
- Make a distinction between global focus and local focus,
- Describe global and local focus theories based on different aspects such as domain structure, dialogue intentions, and semantic relations,
- Describe applications making use of the notion of focus.

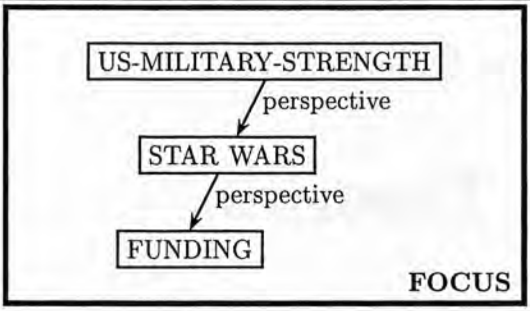
2.1 Introduction

The notion of focus has been studied by the natural language processing community since the 1970s. This notion is concerned with defining what is important in a discourse. It is now recognised as a fundamental aspect of discourse along with the informational aspect (what a discourse is about) and intentional aspect (why things are said in a discourse) (Moore and Pollack 1992). Numerous competing theories have been proposed to explain this notion and research is still very active. Applications using the notion of focus to improve the quality of the text they produce have appeared since the late 1980s. More recent applications (our system is one of them) have investigated the use of the notion of focus in interactive systems where users play a major role in the human-computer dialogue. In this chapter, we present all major focus theories (Lecœuche et al. 1998a). We pay particular attention to explaining in what respect they can, or cannot, be re-used for our purposes.

2.2 Definition

We define the focus as being *the set of all the things to which participants in a dialogue are attending at a certain point in a dialogue and the point of view they have on these things.* The things represent *what* is focused on. They can be concrete or abstract and they can either be explicitly mentioned in the dialogue or they can be implicit. The point of view represents *how* the things are focused on. It depends on the dialogue history.

Figure 2.1 Focus definition (adapted from McCoy and Cheng (1991), p. 109)



For example, imagine a conversation where the US military strength in general is discussed. The conversation then evolves to the Star Wars project and its funding. The focus of this conversation could be represented as in figure 2.1. The things in focus are what has been discussed while the point of view depends on the dialogue history. Here, the funding is seen from the Star Wars project perspective and is related to the overall issue of the US military strength. The perspective would be different if we were considering the Star Wars project as a technological experiment in decision making for example.

Focus represents the conversational context in which future utterances are interpreted. If new sentences cannot be interpreted with respect to the focus, the discourse will seem incoherent. A dialogue can be coherent or incoherent at several levels. For example, the dialogue in example 2.1 is incoherent at the sentence level: the two sentences do not seem to focus on the same topic. In example 2.2, the dialogue is incoherent at a higher level: the

Example 2.1 Sentence coherence (from Schank (1977), p. 421)

A: I just bought a new hat.
B: Fred eats hamburgers.

last paragraph, although internally coherent, does not fit with the preceding conversation.

These different levels of coherence are reflected in the distinction between local focus and global focus (McKeown 1985a, p. 56). On the one hand, *local focus*, also called immediate focus, deals with sentence level coherence (Sidner 1983, p. 107). It explains how two consecutive sentences are related. *Global focus*, on the other hand, deals with higher-level coherence (Grosz 1977, p. 5). It explains how parts of the dialogue are related. As we shall see, distinguishing between these two kinds of focus is necessary because they obey different sets of rules.

2.3 Global focus

In this section, we describe theories concerned with global focus. We classify these focus theories according to the kind of knowledge they use to keep track of the focus. We distin-

Example 2.2 Paragraph coherence (from Sidner (1983), p. 111–112 and 127–128)

B: How do you teach your students to use a calculator?
A: I think students should use a calculator for a while. I give them problems to solve with it, and when they have trouble, I answer their questions about the problems.
B: That's all well and good, but I think they need more instruction on the device to reduce the number of questions. Instead I give them instructions, and they use these to solve problems. They don't have much trouble learning to use the machine.
C: Well, I think you are both wrong. Here's why. I'm going on a vacation to Tahiti tomorrow. I'm going by plane, and I'll be there about a week. It is going to cost me a bundle of money.

guish between theories based on "real world" properties such as task structure and domain structure, theories based on dialogue participants' properties such as intentions or memory limitations, and theories based on dialogue properties such as dialogue moves.

The theories are presented in order of complexity; later theories usually extending previous ones. For each category, we have selected a main piece of work representing it, preferring work that has been used for implementing dialogue management systems.

2.3.1 Based on task structure: Grosz (1977)

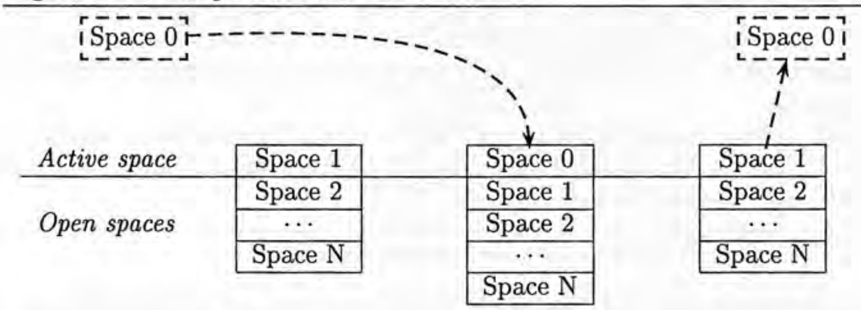
In this approach, the focus is divided into focus spaces organised as a *stack*. A focus space represents the information in focus during part of the dialogue. It is divided into two components. The first component is the *explicit focus*. It contains the objects and events that have been mentioned during that discourse part. The second component is the *implicit focus*. It contains objects and events related to the elements in explicit focus but not directly mentioned in the discourse. Recording both types of elements is important as the dialogue can refer to objects or events in explicit or implicit focus as shown in example 2.3. In this example, the explicit focus only contains references to the lid, the container and the action of attaching them together. However, the bolts are put in implicit focus because they are related to the action of attaching the lid to the container. Therefore, the reference to them is easily understood.

Example 2.3 Implicit focus (from Grosz (1977), p. 68)

S: Attach the lid to the container.
R: Where are the bolts?

The top-most space in the stack is the *active* space. It contains the elements that are in focus at the current point in the dialogue. When a new topic is discussed, a new space is pushed on the stack and therefore becomes active. The other spaces are *open* spaces. They contain information about previous topics that have not yet been completed. This information is still available for reference in the dialogue, but is less accessible than the information in the active space. The open spaces can become active again as the dialogue evolves and shifts back to uncompleted issues. This is shown in figure 2.2. In this example, Space 0 is pushed on top of the existing spaces. It contains elements that are then more salient in the discourse than the elements of the open spaces. Then it is popped and Space 1 becomes active again.

Figure 2.2 Examples of stack movements



Space 0 becomes the new active space. Then it is popped and Space 1 becomes active again.

Grosz used this framework to study the global focus in conversations between an expert and an apprentice assembling part of a compressor. She found that the movements of the focus, i.e., the stacking and un-stacking of focus spaces, mirror the structure of the task discussed (Grosz 1977, p. 104–105). A new focus space is pushed on the stack whenever a new task begins. The focus space is then used for reference during the realisation of the task. It is finally popped from the stack when the task is completed. For example, if completion of Task 1 requires completion of Task 0, a space for Task 1 will first be pushed on the stack. Then a space for Task 0 will be pushed as this subtask is dealt with. Then it will be popped and Task 1 space will become active again. This corresponds to the stack movements in figure 2.2.

The stack movements are reflected in the dialogue by different means. Pushing a new space is usually indicated by a sentence that refers to the goal or the objects involved in the new task. Popping a space is usually marked by references to elements in an open space of the stack. The stack is then popped so that the open space referred to becomes the active focus space again. Recognition of topic shifts is also helped by the task structure *shared* by the participants: the hearer knows when to expect a shift.

As new elements are included into a focus space, an important question is to know what implicit information should be added, since this information can influence focus shifts. Grosz proposes two principles (Grosz 1977, p. 67–68):

- for physical objects, add the subparts of the objects;
- for events, add the sub-events as well as their participants.

However, no reasons are given to explain these principles (see section 2.3.6).

A number of problems remain open in this theory (Grosz 1981). Notably, the relation between focus and belief is not clear. Speaker and hearer may have different knowledge of the concepts spoken about in the dialogue. Therefore, they may focus differently on them (Levelt 1989, p. 119–120). This is not reflected in the theory where a unique focus space is

Table 2.1 Cue words (from Grosz and Sidner (1986), p. 198)

Focus shift	Cue words
dominance push	for example, to wit, first, second, and, moreover, furthermore, therefore, finally
satisfaction-precedes push	in the first place, first, second, finally, moreover, furthermore
pop	anyway, but anyway, in any case, now back to, the end, ok, fine

used. This is because focus is seen as "a property of the discourse itself, not of the discourse participants" (Grosz and Sidner 1986, p. 179). Using two spaces, one for the speaker and one for the hearer, would require one to study issues such as how the two are synchronised and how the speaker's belief about the hearer's focus influences the dialogue. In spite of these shortcomings, the notion of focus space is very important. It is used in numerous global focus theories.

2.3.2 Based on domain structure: Linde (1979)

Linde's approach is very similar to that of Grosz. The focus representation is identical but the shifts depend on the domain structure rather than on the tasks to be carried out (although there is an overlap between the two).

Linde used her framework to study the global focus in descriptions of apartments. She found that the movements of the focus are based on the apartment lay-out. The descriptions begin with the entrance hall and then describe each room in turn as if the speaker was walking through the apartment.

This approach first shows that the task structure is not the only source influencing the focus moves. Domain structure is another possibility. This is confirmed by Levelt (Levelt 1989, p. 142–143). Levelt observed that people describing a simple spatial configuration consisting of linked nodes use it to structure their discourse. In particular, they follow the "principle of connectivity" choosing as the next node to describe one that is linked to the latest node presented. When people have to come back to a previous node because they reach a dead-end, they do it by returning to the nearest choice point not explored. Levelt calls this the "stack principle" which also confirms Linde's theory.

A second point made by Linde is that the task or domain structures are not sufficient to determine the focus moves done in the dialogue. For example, if a room R1 leads to two others, R2 and R3, two possible focus moves are possible: R1 to R2 or R1 to R3. Choosing between these choices depends on criteria independent of the domain structure, such as the intention of the speaker in describing the apartments. For example, in his experiment with a simple lay-out, Levelt found that people tend to minimise their cognitive load by describing simpler parts first, i.e., the ones that require less backtracking.

This theory is of interest to us since organising an elicitation dialogue based on the domain spoken about seems easier than organising it on the task (which can be summarised to "elicit all information you can").

2.3.3 Based on dialogue intentions: Grosz and Sidner (1986)

Grosz and Sidner propose to divide up the dialogue according to the aims of its parts and to base the shifts on this division, rather than on a task or domain structure.

The basic focus mechanism is again similar to that of Grosz's original work. The focus space however not only contains the concepts spoken about but also the "discourse segment purpose" (DSP) (Grosz and Sidner 1986, p. 179). The DSP represents the intention of a particular part of the discourse. DSPs form a structure organised by two relations: dominance and satisfaction-precedes. Dominance means that the completion of one discourse segment contributes to the completion of the dominant discourse segment. Satisfaction-precedes means that the first discourse purpose must be completed before the second can be satisfied. For example, if you intend to open a door, you may first intend me to tell you where the keys are and then you may intend to unlock the door. There is a dominance relation between the intention of opening the door and the intentions of finding the keys and unlocking the door. There is a satisfaction-precedance between the intention of finding the keys and the intention of unlocking the door.

The focus shifts are then dependent on the intentional structure of the dialogue: "a push [on the focus stack] occurs when the DSP of a new segment contributes to the DSP of the immediately preceding segment. When the DSP contributes to some intention higher in the dominance hierarchy, several focus spaces are popped from the stack before the new one is inserted" (Grosz and Sidner 1986, p. 180).

An advantage of this approach is that it bases the focus shifts on *dialogue* intentions and not on structures that are only indirectly related to the dialogue organisation (see sections 2.3.1 and 2.3.2).

A major difficulty with this approach is the need to recognise a speaker's intention to determine the appropriate shift of focus. This implies that the speaker should provide enough information for the hearer to find out the correct focus shift *at the start* of a discourse segment. Three main kinds of information can be used to determine how the DSP of a new segment relates to the previous one: cue words, i.e., special phrases indicating a particular shift, utterance-level intentions and general world knowledge (Grosz and Sidner 1986, p. 188). Since cue words do not require inferences and can be put at the beginning of a new segment, they often indicate focus shifts. Grosz and Sidner propose different cue words for dominance push, satisfaction-precedes push, and pop. (Grosz and Sidner take into account some other relations between discourse segments such as digression or feedback. We do not consider them here.) These cue words are summarised in table 2.1. Recognition of focus shifts thanks to utterance-level intentions or general world knowledge is more complex and not yet fully understood.

Because of the problem associated with recognising participants' intentions, we cannot simply apply this theory for our purposes. However, the idea of indicating focus shifts at the start of every segment by means of cue words can be easily re-used.

2.3.4 Based on memory limitations: Walker (1996b)

Walker points out that stack based theories (see sections 2.3.1, 2.3.2 and 2.3.3) do not include "constraints related to the length, depth, or the amount of processing required for an embedded segment" (Walker 1996b, p. 256). In other words, when a segment is popped from the stack, the previous segment information becomes immediately available again, regardless of the popped segment length. This seems unlikely given people's memory limitations (Miller

Table 2.2 Conversational moves (from Reichman (1984), p. 180)

Conversational move	Cue word(s)
Support	Because...; Like...; Like when...
Restatement and/or conclusion of point being supported	So...
Interruption	Incidentally...; By the way...
Return to previously interrupted context space	Anyway...; In any case...
Indirect challenge	Yes/Right but...; Except, however...
Direct challenge	(No) But...
Subargument concession	All right/Okay but...
Prior logical abstraction	But look/listen/you see...
Further development	Now...

1956). The longer the intervening segment is, the less likely previous segment information will still be available. Therefore, Walker models focus by a cache rather than by a stack. The main characteristics of the cache are (Walker 1993, p. 33+):

1. The elements pushed more recently in the cache are more likely to be retrieved.
2. The elements pushed more frequently in the cache are more likely to be retrieved.
3. Elements are never explicitly popped from the cache but may be displaced. Displaced elements go to main memory. They can later be recalled to the cache if need be.

Because dialogue elements have different importance, priority levels are assigned to them. For example, elements related to the current topic or to a previous point that is not closed are assigned a higher priority level since we expect to continue speaking about them. Elements are then displaced from the cache by reverse priority level order. This explains why we can "keep in mind" information that will be useful when returning to a previous topic (Guindon 1985).

The advantages of Walker's theory over stack based theories are that it provides:

1. A better account of how focused elements are kept salient for some time after their related dialogue segment is closed (Walker 1997). In the stack model, elements are immediately popped.
2. An explanation for "Information Redundant Units" (IRU) (Walker 1996a). These units are so called because they reintroduce information already expressed in the dialogue. Walker argues that one of the roles of these units is to indicate that some elements should be put back into focus, i.e., transferred from main memory to cache. Stack based theories do not offer any explanations for IRU.

However, it is not clear how elements in the cache are prioritised. Recognition of the speaker's intention seems necessary (Walker 1996b). This encounters the same problems as for task based theories (see section 2.3.3).

2.3.5 Based on dialogue moves: Reichman (1985)

This theory has three interesting features: (1) different kinds of focus spaces are distinguished, (2) various dialogue moves are used to navigate between spaces and these moves are constrained by the previous dialogue, and (3) focus spaces have activation levels. We review now these three points in more detail:

1. A first feature of Reichman's work is the use of different focus spaces depending on the nature of the discourse. For example, during debative discourses, `issue` spaces and `support` spaces are used. Differentiating various spaces allows one to record not only the elements in focus and the intention of a discourse part, as in Grosz and Sidner's work, but also specific information related to the particular type of discourse taking place. For instance, an `issue` space lists, among other, what kind of issue (epistemic, evaluative, deontic) is under discussion and its topic. A `support` space on the other hand contains information such as the source of the support, or its credentials. Using a richer representation framework to record specific information about each space allows a finer evaluation of the moves that can take place in the dialogue. It also enables us to interpret some references, such as references to the topic of the focus space, that cannot be interpreted in a simpler framework (Hitzeman and Poesio 1998).

2. A second feature of Reichman's work is the differentiation of several conversational moves to explain shifts in focus. A conversational move is defined as a "communicative act serving a new discourse role" (Reichman 1984, p. 163). In other words, a conversational move provokes a focus shift. Reichman proposes several conversational moves such as interruption and return among others (Reichman 1978, p. 292–298). Each move is defined by a set of preconditions determining when it can apply, and a set of actions defining the result of its application. For example, if a participant defends a claim by making an analogy, a possible move is to challenge this support by expressing doubts on the validity of the analogy mapping. The precondition of this move is that indeed an analogy has been made, and its result is to undermine the claim's support (leading to a counter-challenge or to the concession of the inadequacy of the analogy).

   A set of cue words indicates the moves taken in the dialogue. Some of the conversational moves and their associated cue words are summarised in table 2.2.

3. A final feature of Reichman's work is the recording of a focus space's level of activation. In stack-based theories, the spaces nearer the top of the stack are in higher focus. Spaces not in focus are popped from the stack. Reichman's framework is more complex. Focus spaces are not placed into a stack but assigned a focus level. The focus level has one of the following values (Reichman 1985, p. 54):

   Active indicates the unique space where the current utterances are placed. It has the same meaning as the active space in Grosz's stack framework.

   Controlling indicates the unique space to which the active space is related. Only the elements of the controlling and active spaces can be spoken about in the dialogue.

   Open indicates a previously active space that has not been completed. Open spaces are candidates for focus shifts as in the Grosz stack framework.

   Generating indicates a space in an indirect relation to the active space. For example, an issue space created when a support space is active would become active while the support space would be reassigned a generating focus level.

Closed indicates a space whose conversational move has been completed. Closed focuses are not deleted and may be reopened if necessary (although this is not expected).

Superseded indicates a previously active space that has not been completed but that is not expected to become active again.

Conversational moves modify the focus level of the focus spaces. When a space becomes active, the focus levels of all the focus spaces related to it are reassigned new values depending on their relations with the new active space. For example, an interruption move creates a new focus space, assigns an active level to it and assigns an open level to the previously active space since this space is not related, directly or indirectly, to the new active space (Reichman 1981, p. 52 and p. 100). At the same time, a return move is put into an expectation list. It then has a higher priority to apply than other moves since an interruption should come back to the interrupted space. At the end of the interruption, the expected return move closes the interruption space and re-assigns an active level to the suspended space. The use of the various focus levels as well as the expectation list makes the framework richer than the stack approach.

A possible problem with this approach is the ability to re-open closed focus spaces. As a result, clauses that are far apart in the dialogue can be put into the same space although the context of the dialogue may have changed. Polanyi proposes that closed spaces should remain closed. Related material should be put in a new space which is then explicitly linked with the closed space if need be (Polanyi 1985). The focus evolution is then closer to the dialogue structure (which is also the case for stack-based approaches).

Because of its fine-grained and well-defined nature this theory seems quite appropriate for our purposes. In particular the capability of reopening closed focus spaces is important since people may forget to say things in the elicitation process. In that case, they would need to come back to a previous topic that was considered to have been dealt with.

2.3.6 Summary

Compared with the earlier theories of focus, we can notice in more recent theories an evolution towards richer representations. The representation has been extended along three lines:

- It provides a better account of the role of intention in focusing. This is exemplified in Grosz and Sidner's theory (see section 2.3.3) where the focus shifts are dependent on the discourse participants' intentions.
- It provides a finer description of the dialogue structure. This is exemplified in Reichman's theory (see section 2.3.5) where numerous dialogue moves, with their preconditions and effects, are identified.
- It provides a better model of human information processing. This is exemplified by Walker's theory (see section 2.3.4).

These extensions take into account dialogue properties as well as "real world" properties such as task or domain structures. They are also more realistic by recognising human limitations.

This evolution is accompanied with difficulties. For example, taking intention into account is a difficult task because it implies the capability to recognise speaker's intentions which is a complex problem. To achieve this task, Grosz and Sidner proposes the notion of shared plan between the dialogue participants (Grosz and Sidner 1990; Grosz and Lochbaum 1993). Knowing what has to be done enables them to decide when a shift occurs. The second line of improvement does not encounter such problems. Recognising a dialogue move is usually easier than recognising an intention because dialogue moves are more constrained. Only a few moves can possibly apply at any point in a dialogue. However, while recognition of focus shifts is made easier, the generation of dialogue is not fully specified by these theories. Speaker's intentions should, at some point in the dialogue production process, play a role. Further progress in understanding this problem would require the study of the relation between intentions and dialogue moves in more detail. This relation may be a complex one since intentions and discourse relations are not mapped one-to-one in general (Moore and Paris 1993, p. 667).

A number of issues remain open in the global focus theories (Grosz 1981):

- What is the relation between focus and perspective? People usually focus on concepts from a certain point of view (McCoy 1988). This is not reflected in the focus space where entities are in focus or not. The notion of controlling space is a way of providing some perspective on the current active space. However, it remains a crude description since the relation between controlling and active spaces is not specified. A finer representation is required to reflect this phenomenon. Accordingly, finer shifting rules are required to allow only shifts that are compatible with the perspective taken.

- When is a reference in focus? It is not clear when to consider an inexact referring phrase to still refer to a focused entity or to consider it as a new entity and maybe shift the focus based on that assumption. This problem is similar to deciding what is new or given in a dialogue. Clark and Haviland (1977) propose that a piece of information is considered given if the listener can build "an inferential bridge [to it] from something already known". Some suggestions have been made on how to build such a bridge (Allen 1987, p. 346+) but this topic is still an open issue for research.

On the practical side, theories making use of task stucture or domain structure seem the easiest to program in a computer since representing these structures is quite simple. It remains however to determine which particular aspect, e.g., spatial lay-out, of the domain should be represented. Theories based on intention seem much more difficult to implement in general since recognising intentions in a discourse remains a complex problem not fully understood. These theories may be applicable in limited domains where the number of possible intentions behind a discourse is limited. However, limiting this number too much is detrimental. We could for example reduce the intention of having a requirements elicitation dialogue to a single intention like "express the requirements" but this would not help organise the dialogue.

The choice between stack, cache or focus space activation models is not as important as the preceeding choice of what to model. These three structures are quite well-known and implementing them does not pose big problems. It should be possible to change one for another.

2.4 Local focus

In this subsection, we describe theories of local focus, i.e., theories explaining fine-grained coherence. The theories are classified according to the kind of knowledge they use to keep track of the focus. We again distinguish between theories based on "real world" properties such as domain structure, theories based on participants' properties such as their cultural background, and theories based on dialogue properties such as sentence form and dialogue moves.

2.4.1 Based on grammatical function: Centering (Grosz et al. 1995)

This theory of local focus explains the coherence between consecutive sentences based on the grammatical functions of the concepts in the sentences. Each concept is assigned a focus level depending on its function. These levels then constrain the construction of the next sentence in the dialogue. This sentence sets, in turn, new focus levels that are used to constrain the following one, and the process recurs. We describe this theory in detail as we use it in our system. We formalise it in chapter 3.

Each sentence, except the first sentence of a dialogue, has a main element with respect to the previous sentence. This element is called the backward-looking centre of the sentence. (We write the backward-looking centre of sentence S $Cb(S)$ or Cb for short when there is no ambiguity about the sentence.) The first sentence of a dialogue does not have a backward-looking centre because a sentence *on its own* does not have one. For example, if I say: "I bought a new hat at the shop round the corner yesterday", the sentence could be about me ("I am going to a wedding"), about the hat ("It is nice, isn't it") or about the shop ("It is amazing what they have in the store"). The backward-looking centre is defined by the *relation between two sentences*, i.e., the backward-looking centre of a sentence is a concept which is echoed from a previous one. Each sentence also has a list of forward-looking centres composed of all the concepts present in the sentence. (We write these elements $Cf(S)$ or Cf for short when there is no ambiguity about the sentence.) The list is ordered by decreasing importance. The most important forward-looking centre is called the preferred centre and is expected to become the backward-looking centre of the next sentence. (We write this element $Cp(S)$ or Cp for short when there is no ambiguity about the sentence.)

Example 2.4 The semantic nature of centres (from Grosz et al. (1983, p. 45))

3a	How is Rosa?
3b	Did anyone see her yesterday?
3c	Max saw her.

Centres are semantic entities, not linguitic ones. In example 2.4, the backward-looking centre in 3c refers to Rosa through the intermediary of "her" in 3b. If 3a was changed (say, to speak about Joan) then the backward-looking centre in 3c would refer to the person spoken about (i.e., Joan). Therefore, the Cb is a semantic entity and not a linguistic one (which is still "her" in 3b in both cases).

The current backward-looking centre, when it exists, and the forward-looking centres list are used to determine the backward-looking centre of the next sentence. In other words, if we slide a window spanning two sentences over the dialogue, the first sentence in the window sets expectations on what the backward-looking centre will be, based on the current backward-looking centre and the new concepts introduced, and these expectations are confirmed, or not, by the other sentence.

The forward-looking centres are ordered by the following rules:

- A sentence subject is ranked higher than its objects.
- Objects are ranked higher than the other elements.

Several possible focus transitions between consecutive sentences are also defined (Grosz et al. 1995, p. 210):

Continuation The backward-looking centre of sentence n+1 is the same as the backward-looking centre of sentence n ($Cb(S_{n+1}) = Cb(S_n)$) and is also the preferred centre of sentence n+1 ($Cb(S_{n+1}) = Cp(S_{n+1})$). In that case, it is likely to be the backward-looking centre of sentence n+2.

Retaining The backward-looking centre of sentence n+1 is the same as the backward-looking centre of sentence n ($Cb(S_{n+1}) = Cb(S_n)$) but is not the preferred centre of sentence n+1 ($Cb(S_{n+1}) \neq Cp(S_{n+1})$). In that case it is unlikely to be the backward-looking centre of sentence n+2.

Shifting The backward-looking centre of sentence n is not the same as that of sentence n+1 ($Cb(S_{n+1}) \neq Cb(S_n)$).

These transitions are summarised in table 2.3.

Table 2.3 Transitions in Centering Theory

	$Cb(S_{n+1}) = Cb(S_n)$	$Cb(S_{n+1}) \neq Cb(S_n)$
$Cb(S_{n+1}) = Cp(S_{n+1})$	Continuing	Shifting
$Cb(S_{n+1}) \neq Cp(S_{n+1})$	Retaining	

These transitions and the forward-looking centres are used to select the backward-looking centre of the next sentence by applying two rules (Grosz et al. 1995, p. 214):

Rule 1 If any element of the forward-looking centres of sentence n is realised by a pronoun in sentence n+1, then the backward-looking centre of sentence n+1 must also be realised by a pronoun.

Rule 2 Sequences of continuation transitions should be preferred over sequences of retaining transitions. These sequences should themselves be preferred over sequences of shifting transitions.

These rules constrain what the backward-looking centre can be in several ways. In text understanding for example, rule 1 restricts the backward-looking centre to the elements realised by pronouns (if any). In text generation, rule 2 provides constraints on what can be said next depending on the current backward-looking centre (see section 2.6).

This approach is now applied on a small example (see example 2.5). For each sentence we indicate, when they exist, the backward-looking centre, the forward-looking centres, the preferred centre and the transition move done.

Example 2.5 Centering example (adapted from Grosz et al. (1995), p. 214)

a. John has been acting quite odd.
Cf = {John}, Cp = John
b. He called up Mike yesterday.
Cb = John, Cf = {John,Mike}, Cp = John
c. Mike was studying for his driver's test.
Cb = Mike, Cf = {Mike, driver's test}, Cp = Mike; Shifting
d. He was annoyed by John's call.
Cb = Mike, Cf ={Mike, call, John}, Cp = Mike; Continuation

Table 2.4 Transitions in Centering Theory Extension

	$Cb(S_{n+1}) = Cb(S_n)$	$Cb(S_{n+1}) \neq Cb(S_n)$
$Cb(S_{n+1}) = Cp(S_{n+1})$	Continuing	Smooth-Shift
$Cb(S_{n+1}) \neq Cp(S_{n+1})$	Retaining	Rough-Shift

Extension of the Centering Theory An extension of the centering theory has been proposed in Brennan et al. (1987) by remarking that the shifting transition can be refined into two transitions:

- A smooth-shift transition holds when the backward-looking centre of sentence n+1 is not the same as the backward-looking centre of sentence n ($Cb(S_{n+1}) \neq Cb(S_n)$) but is the preferred centre of sentence n+1 ($Cb(S_{n+1}) = Cp(S_{n+1})$).
- A rough-shift transition holds when the backward-looking centre of sentence n+1 is not the same as the backward-looking centre of sentence n ($Cb(S_{n+1}) \neq Cb(S_n)$) and is not the preferred centre of sentence n+1 either ($Cb(S_{n+1}) \neq Cp(S_{n+1})$).

These transitions are summarised in table 2.4.

The transition ranking becomes (see rule 2 of centering theory): Continuing > Retaining > Smooth-Shift > Rough-Shift. Making this finer distinction is useful for the resolution of pronoun references in some dialogues. For example, centering theory cannot predict the references in the last sentence of example 2.6, but the extended theory can (see example 2.7). The finer distinction also provides finer constraints for text generation.

Example 2.6 Centering example (adapted from Brennan et al. (1987), p. 157)

Brennan drives an Alfa Romeo.
Cf = {Brennan, Alfa Romeo}, Cp = Brennan
She drives too fast.
Cb = Brennan, Cf = {Brennan}, Cp = Brennan; Continuing, she = Brennan
Friedman races her on weekends.
Cb = Brennan, Cf = {Friedman, Brennan, weekends}, Cp = Friedman; Retaining, her = Brennan
She often beats her.
Cb = Friedman, Cf = {?, ?}, Cp = ?; Shifting, she = ?, her = ?

Example 2.7 Extended centering example (from Brennan et al. (1987), p. 157)

Brennan drives an Alfa Romeo.
Cf = {Brennan, Alfa Romeo}, Cp = Brennan
She drives too fast.
Cb = Brennan, Cf = {Brennan}, Cp = Brennan; Continuing, she = Brennan
Friedman races her on weekends.
Cb = Brennan, Cf = {Friedman, Brennan, weekends}, Cp = Friedman; Retaining, her = Brennan
She often beats her.
Cb = Friedman, Cf = {Friedman, Brennan}, Cp = Friedman; Smooth-Shift, she = Friedman, her = Brennan

There are a number of remaining problems with centering theory (Kehler 1997; Walker et al. 1997c):

- As it is purely based on sentence form, semantic incoherence is not detected. Example 2.1 (see section 2.2) is not considered as being particularly incoherent. Detecting semantic incoherence would require one to reason about the meaning of the sentences. A solution is to add a reasoning module with the sole purpose of accepting or rejecting the propositions made by the focusing algorithm (thus having a limited search space to explore), but such a module has not yet been implemented (Sidner 1979).
- Since the theory takes into account only a limited amount of information on the dialogue, e.g., backward-looking and forward-looking centres, some dialogues appear equivalent for the theory but in fact are not. In that case, wrong predictions about the focus are made. This is the case in example 2.8. Although sentences 4a and 4b are equivalent for the theory since they follow the same previous dialogue and have a similar structure, the pronoun used in 4a refers to Bob Dole whereas the one used in 4b refers to Bill Clinton. In both cases the theory proposes Bob Dole as the preferred pronoun.

Example 2.8 A problem with centering (from Kehler (1997), p. 473)

1	The three candidates had a debate today.
2	Bob Dole began by bashing Bill Clinton. *Cf = {Bob Dole, Bill Clinton}, Cp = Bob Dole*
3	He criticized him on his opposition to tobacco. *Cb = Bob Dole, Cf = {Bob Dole, Bill Clinton}, Cp = Bob Dole, he = Bob Dole, him = Bill Clinton*
4a	Then Ross Perot reminded him that most Americans are also anti-tobacco. *Cb = Bob Dole, Cf = {Ross Perot, Bob Dole}, Cp = Ross Perot, him = Bob Dole*
4b	Then Ross Perot slammed him on his tax policies. *Cb = Bob Dole, Cf = {Ross Perot, Bob Dole}, Cp = Ross Perot, him = Bob Dole (?)*

Table 2.5 Focus levels (from Reichman (1981), p. 120)

Mode	Focus Level	Example
Pronominal	High	Your having called him up.
Name	Medium	Your having called Mark up.
Description	Low	Your having called your son up.
Implicit	Zero	Your having called.

- Because the theory applies at the level of the sentence, it cannot be applied incrementally. As a result, intra-sentential dependencies are difficult to take into account. (An incremental model is proposed for example in Strube (1998). However, this proposal is still too recent to know if it can deal with the same range of dialogue phenomena as centering does.)

- It is not clear that the ranking of the forward-looking centers should be done on purely grammatical grounds. Taking the thematic role of elements in the sentence, e.g., actor, experiencer, object (Fillmore 1968), may be more adequate, especially for languages where the role of elements and their grammatical functions are not the same as in English (Strube and Hahn 1996; Walker et al. 1990).

- The relation between global and local focus is not clearly explained by the theory (Grosz and Sidner 1995). The consequences of global shifts on the local focus remain unknown. An interesting attempt at bridging the gap between local and global focus is made in Hahn and Strube (1997). In that approach, global focus spaces are made of sentences with the same preferred centre. Local focus shifts can then be used to detect global focus moves. Although this approach seems to be useful for anaphora resolution, the notion of global focus it adopts is somewhat more limited than the usual one, where global focus groups possibly many different things as long as they are closely related. Nevertheless, this theory points out interesting results. For example, local continuation shifts should be associated with global no change moves since they continue to refer to the same things. Sentences without backward-looking centers are often associated with global pop moves since they do not refer to things just discussed in the dialogue. Some of these results can be observed in our system when the global and local focus theories are both used to organise the dialogue (see section 7.4).

Some of these problems are partly tackled by the following theories.

Because centering is the major theory explaining local focus and since it is already well formalised, it seems a good base for a formal local focus theory for our system.

2.4.2 Based on dialogue moves: Reichman (1985)

This theory completes Reichman's global focus approach (see section 2.3.5). Each focus space, which contains all the elements in global focus, is divided into four categories. These categories correspond to different levels of local focus. Depending on these levels, entities will have different realisations. The levels are described in table 2.5, along with their effect on realisation (called mode in the table). The rules governing the attribution of focus level to the different entities are based on those of Sidner (1979). Here are the most important (Reichman 1981, p. 73):

Table 2.6 Conversation/associational categories (from Schank (1977), p. 433 and 438)

COCAINE				
	FUNCTION	get high	RESULT	get stoned
	PRICE	middle	ENABLE	find a dealer
	OWNER	nil	USER	nil
	AVAILABILITY	hard to get	PROBLEM	illegal
BUY				
	SCRIPT	shopping	ASSOC	nil
	RESULT	own	ACTOR	nil
	EFFECT	nil	OBJ	nil
	FREQUENCY	high	UNUSUAL	2

- The subject of a sentence, of a there-insertion[1] clause, of a cleft, pseudo-cleft or topicalised clause[2], is assigned a high focus level.
- A non subject referenced by name is assigned a medium focus level.
- A non subject referenced by description is assigned a low focus level.
- An entity referenced by name after a previous reference by description is assigned a high focus level.
- If an entity's high focus level assignment is usurped by another entity then the old high focus constituent is reassigned a medium focus level.

These rules are based on grammatical functions as in the centering theory. An important difference between this theory and the centering theory though is the recognition of the global focus shifts' influence on the local focus. Changes in global focus level cause local focus changes. For example, "initial focus level assignments to entities in a digression space, which were previously mentioned in the context that this space interrupted, are carried over from that space" (Reichman 1985, p. 75) since the entities are still fresh in the mind of the dialogue participants. On the other hand, closing a space results in "the removal of all elements from focus, which is reflected in their reassignment to a zero focus level" (Reichman 1985, p. 82) since the elements in a closed focus space are not supposed to be spoken about again. This theory cannot however explain why some local phenomena cross global focus boundaries (Walker 1997).

2.4.3 Based on semantics: Schank (1977)

The theory presented by Schank deals with semantic coherence between sentences. It is based on the dialogue participants' perception of the relations between topics.

As in the centering theory, the focus is defined by a relation between two sentences. A first sentence defines a focus set composed of the objects, persons, locations, actions, states

[1] A "there-insertion" sentence has the following form : *There is* NP. For example, "There is a research group that works on knowledge-based systems" is a there-insertion sentence.

[2] A "topicalised" clause is a clause in which one constituent has been placed unexpectedly at the start. For example, "To John I gave the strawberries" is a topicalised clause with topic "John".

and time mentioned in it. A second sentence modifies that set in two ways. First, a subset of the initial set, called the reduced old topic, is extracted. It represents the concepts that are common to the two sentences. Second, a new set is created. This second set, called the new topic, represents the new concepts that have been introduced in the second sentence. The speaker then has a choice, when continuing the discourse, to pick the new focus in either set. The speaker can maintain the old focus by choosing to discuss an element of the reduced old topic or shift the conversation to an element of the new topic. Even when maintaining the old focus, the speaker will have to acknowledge the new topic, since ignoring it would certainly be considered as very impolite. The acknowledgement can been done for example by rephrasing the old focus from the new focus point of view. In all cases, the new sentence provides a list of foci that will be used to redefine the reduced old focus and new topic. The discourse can therefore be modelled by a binary tree where each node represents a sentence and the two branches represent the maintain focus choice and the focus shift choice respectively. If the conversation runs into problems, e.g., the topic becomes uninteresting, a speaker can jump back to a previous point of choice in the discourse and develop another branch of the tree. Such moves are usually marked in the dialogue by cue words (Schank 1977, p. 428).

Example 2.9 Coherent dialogue (from Schank (1977), p. 432)

> I just got some cocaine.
> Where did you find a dealer?

The main issue of the Schank theory is the process of selecting the new possible topics based on the reduced old topic, i.e., what are the shifts allowed in the dialogue? Schank proposes to associate conversation / associational categories with each concept. These categories describe the elements that are related to a concept and their level of interest. For example, "cocaine" and "buy" have the conversation / associational categories given in table 2.6. These categories can be viewed as putting interesting items in implicit focus (see section 2.3.1). Based on these categories, the speaker can select an interesting shift in the conversation. In example 2.9, the speaker realises that a major problem in having some cocaine is to get it (the value of the availability category is "hard to get"). Therefore, an interesting question is to find out where a dealer can be found ("find a dealer" being the value of the enable category).

On the other hand, example 2.10 seems odd since the actor of the buy action is not important. To reflect the fact that actors of buy actions are usually not important, the value of the actor category for buy is nil (although, in some cases, the buyer may be important such as in the following sentence: "Bill Clinton bought a new car.").

Example 2.10 Incoherent dialogue (from Schank (1977), p. 439)

> I just bought a new car.
> I know someone else who buys things.

The main problem with this theory is the necessity to have conversation / associational categories for each possible concept that can be spoken about in a discourse. Moreover, the value of these categories may vary depending on the participants cultural background or

Table 2.7 Candidate focus shifts (from McCoy and Cheng (1991), p. 112)

Type	Focus shift candidates
Object	• attributes of the object, • actions the object plays a prominent role in (e.g., is actor of)
Attribute	• objects which have the attribute, more specific attribute
Setting	• objects involved in the setting
Action	• actions which typically occur in the setting • actor, object, etc., of the action – any participant role • purpose (goal) of action, next action in some sequence, sub-actions, specialisations of the action
Event	• actions which can be grouped together into the event

the dialogue's context for example. This makes it very difficult to use this theory for our purposes.

2.4.4 Based on domain structure: McCoy and Cheng (1991)

The theory of McCoy and Cheng is somewhat similar to that of Schank. A first sentence puts an element into focus. Depending on the nature of that element (object, action, etc.) certain focus shifts are allowed as shown in table 2.7. The speaker selects one of these shifts and puts a new element in focus. Depending on the nature of this new element and on the previous dialogue history, new shifts will be allowed. This process continues for the rest of the discourse. In this approach, the foci are organised as a tree, called the *focus tree*: nodes represent the focus of sentences and arcs represent shifts between them. The topic at the current point in the dialogue is the list of foci from the node representing the current sentence up to the root node of the tree. That is to say: the topic is the current focus seen in the perspective of the preceding dialogue. As in Schank's theory, the speaker has the possibility to stop developing the current node and instead expand a previous node. In that case, the speaker will have to mark this move in the dialogue so that hearer can understand how the discourse is evolving. The mark needs to be more and more explicit as the dialogue deviates from its normal flow. Such marks may be cue words, tense shift, anaphora usage or pronoun shifting. McCoy and Cheng do not explain how cue words should be chosen for marking shifts.

2.4.5 Summary

Three classes of theories can be distinguished from the presentation:

- Those based on dialogue moves and sentence construction.
- Those based on dialogue participant's beliefs.
- Those based on real world organisation.

Each kind of theory is incomplete by itself: for example, current theories based on sentence form need to be complemented by some semantic checker (see section 2.4.1) and theories based on semantic relations do not explain how the sentences are organised. Bridging the

Table 2.8 Distinction between global focus and local focus

	Global focus	Local Focus
Deal with	High-level coherence	Consecutive sentence coherence
Contain	Several elements	One element
Shifts marked by	Cue words and Anaphora	Anaphora

gap between the theories seems necessary to obtain a general theory of focus. Another issue that remains largely unexplained is the relation between local and global focus. Reichman's theory as well as Hahn and Strube's theory are interesting attempts at integrating these. However, more work needs to be done to understand the influence between these two phenomena. Finally the relation between focus and belief is still a major problem because it requires participants to share the same frame of reference.

On the practical side, theories based on grammatical functions or domain structure seem relatively straightforward to implement. Theories based on semantics on the other hand are much more difficult to program efficiently without the need for an extensive common-sense and cultural knowledge. Since the relation between the different focus theories is still not fully understood, using theories for local focus and global focus based on the same underlying model, such as domain structure, could be risky. Some phenomena may not be accounted by those theories. It may be preferable to use orthogonal theories, such as a local one of those based on grammatical functions which have no equivalent at the global focus level, to reduce such risks.

2.5 Global focus and local focus revisited

From the preceding presentations, we can notice several differences between local and global focus. They reflect the fact that the local focus is concerned with consecutive sentence coherence, while the global focus is concerned with higher level coherence. In particular:

- At any point, only one thing can be in local focus. On the other hand, the global focus may contain several things.
- The local focus evolution is marked in the dialogue by linguistic phenomena such as anaphora. Evolution of the global focus is usually marked in the dialogue by cue words. Anaphora can also be used to mark a global shift since reference to a concept in a stacked focus space can cause this space to become active again.

The distinction between global and local focus is summarised in table 2.8.

2.6 Applications

In this section we describe some practical applications of the notion of focus. These applications are related to our research on dialogue in requirements elicitation systems. In this kind of dialogue, it is of the utmost importance to keep users interested in the dialogue while presenting what the system is doing. Using focus theories, it is possible to know what information should, or should not, be presented and when to present it.

Example 2.11 Centering example (adapted from Grosz et al. (1995), p. 214)

a. John has been acting quite odd.
b. He called up Mike yesterday.
c. Mike was studying for his driver's test.
d. He was annoyed by John's call.

For example, local focus theories (see section 2.4.1) can be used to improve text generation. Here are two examples demonstrating this.

Consider again example 2.5 (repeated here as example 2.11 for convenience). Suppose that the first two sentences have already been produced and that we want to say that Mike was studying for his driver's test. Should "Mike" be realised by a pronoun or not, i.e., should we use sentence c as it is in example 2.5 or can we say "*He* was studying for his driver's test"? If we chose the latter option, rule 1 of the centering theory would put a preference on interpreting "he" as being the focus of sentence b, i.e., John. Therefore, the realisation of Mike by a pronoun is impossible.

Now suppose that only the first sentence has already been produced and that we want to add to sentences b and c that John has been working very hard recently. How should we organise the text? Since we want to speak about Mike in sentence c and the only sentence having Mike in its forward-looking centres is sentence b, these two sentences should stay together. As a result, we have to introduce our new information immediately, producing the following text:

a. John has been acting quite odd.
a'. He has been working hard recently and is quite tired.
b. He called up Mike yesterday.
c. Mike was studying for his driver's test.

Focus theories have been used in various applications. In this section we first present two applications where the notion of focus is applied to the production of multi-sentential explanations. For this kind of explanation, focus rules need to be applied to produce coherent texts of high-quality. Otherwise, the explanations are no better than a series of disconnected sentence-long utterances. The first application is used to present information stored in a database. The system makes use of focus rules to improve the organisation of the text. The second application is used to present arguments in a coherent way by planning the text it produces. The planning is influenced by focus goals and operators. These two applications are interesting for us since we need to organise the elicitation dialogue as they organise the presentation of information. Rather than presenting things however, we ask questions about the specification. Then, we describe the application of the notion of focus to create collaborative agents with dialogue processing capabilities. These agents are used to help system users carry out tasks. This application shows that the focus theories can be applied to more interactive domains than the production of explanations, and is therefore of particular interest to us.

2.6.1 Based on schemata: McKeown (1985b)

The aim of this application is to answer users' questions about the content of a database. An example of an answer generated by the system is given in table 2.9. The system uses

Table 2.9 "What is a ship?" (from McKeown (1985a), p. 30)

What is a ship?
A ship is a water-going vehicle that travels on the surface. Its surface-going capabilities are provided by the DB attributes DISPLACEMENT and DRAFT. Other DB attributes of the ship include MAXIMUM SPEED, PROPULSION, FUEL (FUEL CAPACITY and FUEL TYPE), DIMENSIONS, SPEED DEPENDENT RANGE and OFFICIAL NAME. The DOWNES, for example, has MAXIMUM SPEED of 29, PROPULSION of STMTURGRD, FUEL of 810 (FUEL CAPACITY) and BNKR (FUEL TYPE), DIMENSIONS of 25 (DRAFT), 46 (BEAM), and 438 (LENGTH) and SPEED DEPENDENT RANGE of 4200 (ECONOMIC RANGE) and 2200 (ENDURANCE RANGE).

predefined schemata to output text. These schemata are complemented by global focus and local focus rules to improve the quality of the output.

More precisely, the system starts by selecting a set of schemata that can be used to answer the question of the user. Each schema corresponds to a strategy to answer the question. Examples of schemata are given in table 2.10.

At the same time, the system collects all the knowledge it has on the particular topic of the question by querying its database. Finally, the system finishes this first step by selecting a unique schema from the set of schemata based on the available information from the database. Therefore, at the end of this step, the system has selected a strategy to present its answer and it has partitioned the knowledge that will be used for this presentation, i.e., it has set the global focus of the answer.

The second step consists of actually producing the answer in a coherent way, i.e., in respecting local focus constraints. In each schema, choice points enable the system to vary its presentation. Therefore, the system is able to select a presentation based on the available knowledge and on the preceding utterances. The rules used by McKeown to make this selection are the following:

- Changing the current focus to a forward-looking centre as defined in the centering theory (see section 2.4.1) should be preferred to maintaining the focus. Indeed, if the forward-looking centre is not presented immediately, the system would have to re-introduce it at a later point, which might be difficult (McKeown 1985b, p. 64).
- Maintaining the current focus should be preferred to shifting back to a previous focus. Indeed, it is better to present a topic completely before going back to a previous one, otherwise the topic would have to be re-introduced (McKeown 1985b, p. 66).
- If several shifts are still possible, select the new topic which is most semantically related to the current focus (McKeown 1985b, p. 67).

Table 2.10 Examples of schemata (from McKeown (1985a))

schema	use
identification	used to provide a definition
attributive	used to illustrate a particular point about a concept
compare and contrast	used to describe something by contrasting a positive point against a negative one

Table 2.11 Partial identification schema (based on McKeown (1985a), p. 12 and 19–20)

Identification schema
Identification (class & attribute/function) {Analogy/Constituency/Attributive}* Particular-illustration/Evidence+ ...
'{}' indicates optionality, '/' indicates alternatives, '+' indicates that the item may appear 1 to n times and '*' indicates that the item is optional and may appear 0 to n times.

These rules are underspecified since they may not always select a unique topic. In this case McKeown's system selects a by-default topic which is predefined for each communication. (This is a way of automatically choosing the preferred centre of a sentence.) This rule, which is a catch-all rule applied when the others fail, could be refined by more specific rules based for example on pragmatics such as: select topics with good affect if we want the hearer to feel good or topics with bad affect if not (Hovy 1988a, p. 72). This would enable the system to discriminate communications better. However, for its current purpose of presenting database information, the rules proposed are sufficient.

We now show on the text of table 2.9 how the two steps apply. Suppose that the user asks what is a ship. The system, based on its available knowledge and presentation strategies, selects the identification schema. At the same time, the knowledge that is relevant to the schema is selected from the database and puts it in a "knowledge pool". This sets the global focus of the explanation since only the information in the knowledge pool can be presented. The first two actions of the schema are the identification of the concept to explain and the choice between providing an analogy, a constituency, an attributive, a piece of evidence or a particular-illustration phrase (see table 2.11). The first action results in the first sentence of the answer describing a ship as a water-going vehicle. To carry out the second action, the system must apply the focus constraints since several focus shifts are possible. The particular-illustration move is ruled out because it requires information not included in the relevant knowledge pool. To choose between the remaining shifts, the system applies the local focus rules. All the available moves would lead to focus on the ship except for the evidence move which makes information on the "surface-going capabilities" salient[3]. Since the surface-going capabilities are a member of the forward-looking centres of the current sentence, this shift is preferred to the others and the system describes these capabilities in the second sentence of the answer. The system then goes through the rest of the schema applying the focus rules at all the choice points.

Another approach, somewhat similar to this one, is proposed in Carcagno and Iordanskaja (1993). In that approach, the notion of schema is played by a "focus tree". The focus tree represents what can be spoken about. It is then filled with information retrieved from a database. Transformation operations group similar parts of the tree together. Finally, the tree is used to output text.

[3]For each fact that could be chosen by a move, a default focus indicates what is most likely to be focused on.

2.6.2 Based on domain structure: Reed and Long (1997a)

This approach is concerned with planning argument presentations in natural language. The order in which arguments are presented is important since it influences their persuasive effect and their coherence. Planning text is an active area of research in natural language generation (Hovy 1988b, 1991, 1993; Moore and Swartout 1989; Moore and Pollack 1992; Moore and Paris 1993; Young and Moore 1994; Young et al. 1994). However, very few approaches take focus into account. Most approaches are based on communicative goals which define what the system is trying to achieve and planning operators which are used to achieve these goals. The main issue is then to find an acceptable way of applying the planning operators to achieve particular communicative goals. In the approach presented here, some communicative goals and planning operators are used to manipulate focus explicitely. They are:

is_salient(Hearer, Proposition, Context) This is a communicative goal which means that Proposition must be in focus for Hearer within Context. (Context plays the role of a focus space.)

make_salient(Hearer, Proposition,Context) This is a planning operator which puts Proposition in focus for Hearer within Context. The way this is done depends on the realisation of the discourse plan by an Eloquence Generation (EG) module producing the actual text.

push_topic(Context) and **pop_topic(Context)** These two communicative goals are used to structure the argument. Context is used as a focus space. These two goals give the system a stack-based focus approach. They can be realised by the EG module by means of cue words, punctuation or formatting.

Using these basic goals and operators, we can define, for example, how to present a modus ponens argument as shown in table 2.12 (`bel(Hearer, Proposition)` is the communicative goal of having Hearer believe Proposition). The presentation is divided into four main parts:

t_0 Create a new context in which we will speak about Proposition,

t_1 **and** t_2 Persuade the hearer of the validity of a new proposition X and put this proposition in focus,

t_3 **and** t_4 Persuade the hearer of the validity of the implication relation between X and Proposition and put the implication in focus,

t_5 Conclude about the validity of Proposition and pop the context.

Some reordering of the goals and operators may be required to ensure the coherence of a text (Reed and Long 1997b). For example, if the conclusion of an argument is supported by a very long chain of deduction, it may be better to present the conclusion first and the support second so that people know what the system is trying to achieve. On the other hand, if the support is short, there is no need to do this.

Focus operators have also been used in some other approaches, in particular general text planning (Hovy and McCoy 1989) and theorem proof presentations (Huang 1994a,b; Huang and Fiedler 1997). In each case, the focus operators ensure that the text not only achieves its persuasive aim but also that it is coherent and easy to understand.

Table 2.12 Modus Ponens operator (from Reed et al. (1997))

Modus Ponens (Hearer, Proposition)		
Body	t_0	push_topic(Proposition)
	t_1	bel(Hearer, X)
	t_2	is_salient(Hearer, X, Proposition)
	t_3	bel(Hearer, X $\supset$ Proposition)
	t_4	is_salient(Hearer, X $\supset$ Proposition, Proposition)
	t_5	pop_topic(Proposition)

2.6.3 Based on dialogue intentions: Collagen (Rich and Sidner 1997)

Collagen is an application-independent toolkit used to implement collaborative interface agents with discourse processing capabilities. These agents help the users of a system by conversing with them and studying their actions. The agents can act themselves directly on the system. They have access to a database of application-dependent plans that can be applied to the system to carry out some goals. We focus here on their dialogue capabilities.

Collagen agents decompose a dialogue into segments according to the Grosz and Sidner global focus theory (see section 2.3.3). Each segment is therefore associated with its intention and the entities in focus during its realisation. Segments are put into a stack when they begin and are popped when they finish. Based on this decomposition and on the plans they have access to, agents propose actions to be performed. To carry out this task, the discourse generation algorithm produces an agenda of prioritised actions that would contribute to the current discourse segment purpose (DSP). The priority is computed thanks to the focus of attention (among other factors). For example if objects used in the action considered are already known, this action will receive a high-priority. Collagen agents therefore ensure that the action proposed is always coherent with the current user preoccupations.

Collagen agents also make use of this mechanism to reduce the need for natural language understanding. Rather than tackling unrestricted natural language inputs, the agents propose a dynamic set of inputs to the users. This set is computed from the current discourse context and the possible actions that the user can do in the given situation. The agent-user interactions are therefore adapted to the current needs of the user and are easy to understand since they follow dialogue rules. This approach, although less powerful than having a fully-fledged natural language understanding module, seems cost-effective and well-adpated with the idea of having a focus toolkit ready to be used in a variety of applications.

An example of agent-user communication is given in table 2.13. The agent and the user are trying to schedule a trip using an application displaying a map and computing the possible itineraries. (Remember that users select their input from a set of predefined sentences proposed by the system.)

Collagen agents have some similarities with the approach we present in the thesis. However, the aim of our project is different from the goal in Collagen. Collagen agents are used to give help to users while working with an existing and fully-fledged application. Our aim is to provide an application with a flexible natural language interface. The main differences between Collagen and the system we propose are the following:

- We do not allow users to have direct access to the requirements elicitation tool. All interactions are processed by the dialogue manager first. This is a limitation on the

Table 2.13 Collagen agent dialogue (adapted from Rich and Sidner (1997))

USER: Propose scheduling a trip.
AGENT: OK.
(User identifying route of scheduling a trip as San Francisco to Dallas to Boston)
USER: Add San Francisco to the route.
USER: Add Dallas to the route, allowing 97 itineraries.
USER: Add Boston to the route, allowing 100+ itineraries.
AGENT: How should a trip on the route be scheduled?
USER: Propose scheduling on the route via working forward.
USER: What should be done next?
AGENT: Propose you work on San Francisco to Dallas leg.
...

users freedom but it enables us to keep a better record of what happens. It makes also a clearer separation between the program in charge of interacting with users and the program in charge of reasoning about the requirements.

- We do not maintain a plan library. There may exist many ways of carrying out the elicitation task. It would therefore be difficult to store predefined ways of working.
- Our system is able to use several focus theories and to combine their recommendations to choose the best way of continuing a dialogue.

2.7 Summary

We can summarise the main points of the three examples of application we presented as follows: In the non-interactive applications, global focus is mainly used to structure the dialogue a priori, e.g., by partitioning the knowledge that will be used to answer a question, or by using push and pop operators. Local focus is then used to fine-tune the text by taking into account what has been said previously. In the interactive applications, the a priori structuring phase cannot be carried out since the dialogue organisation depends on what users will say. However, it is still possible, in general, to recognise different levels of focus (although in Collagen only the global focus based on intention is tracked). In this thesis, we propose a system which can track local and global focus in an interactive discourse. Given the state of the art in natural language understanding and user modelling, it seems difficult to apply some of the focus theories based on intentions or semantics. We will therefore concentrate on theories based on task, domain and sentence structures since they are easier to examine.

Table 2.13 Collagen agent dialogue (adapted from Rich and Sidner (1997)).

USER: Propose scheduling a trip.
AGENT: OK.
(User identifying route of scheduling a trip as San Francisco to Dallas to Boston)
USER: Add San Francisco to the route.
USER: Add Dallas to the route, allowing 97 itineraries.
USER: Add Boston to the route, allowing 100+ itineraries.
AGENT: How should a trip on the route be scheduled?
USER: Propose scheduling on the route via working forward.
USER: What should be done next?
AGENT: Propose you work on San Francisco to Dallas leg.

users freedom but it enables us to keep a better record of what happens. Besides this, there is a separation between the program in charge of interacting with users and the program in charge of reasoning about the requirements.

- We do not maintain a plan library. There may exist many ways of carrying out the elicitation task. It would therefore be difficult to store predefined ways of working.

- Our system is able to use several focus theories and to combine their recommendations to choose the best way of continuing a dialogue.

2.7 Summary

We can summarize the main points of the three examples of application we presented as follows. In the non-interactive applications, global focus is mainly used to structure the dialogue a priori, e.g. by partitioning the knowledge that will be used to answer a question, or by using push and pop operators. Local focus is then used to determine the text by taking into account what has been said previously. In the interactive applications, the a priori structuring phase cannot be carried out since the dialogue organization depends on what users will say. However, it is still possible, in general, to recognize different levels of focus (although in Collagen only the global focus based on intention is tracked). In this thesis, we propose a system which can track local and global focus in an interactive dialogue. Given the state of the art in natural language understanding and user modelling, it seems difficult to apply some of the focus theories based on intentions or semantics. We will therefore concentrate on theories based on task, domain and syntactic structures since they are easier to examine.

Chapter 3

Formal Theories of Dialogue Coherence

Objectives

The objectives of this chapter are to:

- Present the formalisation of two focus theories in the form of rules and orderings on these,
- Present the need for "translation" rules between the application domain and the focus theories,
- Present the formalisation of other theories needed in the handling of dialogues.

3.1 Introduction

Most of the focus theories presented in chapter 2 suffer from one of two problems. The first problem is that they are not formal enough to be easily written as computer programs. This is, for example, the case of all the theories where semantic relations between phrases must be taken into account to evaluate their coherence. It is not clear how such relations can be computed. Using WordNet (Fellbaum 1990; Gross and Miller 1990; Miller 1990; Miller et al. 1990) could be a possibility since words are grouped into coherent sets based, for example, on hypernymy and synonymy. However, it is not obvious how these groups are related to the relations used in the focus theories. The second problem of some focus theories is that they have been designed and formalised for natural language understanding and in particular anaphora resolution. This is particularly true for the leading local focus theories. Using them for text generation has been largely ignored except for a few practical pieces of work (Mittal et al. 1998). In this chapter, we present a formalisation of two focus theories. A global focus theory is first presented. It has been written so that it can easily be implemented and used for text generation. The relation between the notions used to formalise the theory and the domain knowledge likely to be found in the application domain of the text generator is analysed and the need for "translation" rules between the two sets of notions is discussed.

(The rules themselves will be presented in more detail in chapter 4 where the representation used by the elicitation system is discussed.) A local focus theory is also presented. It has again been written to be easily implemented in a text generator. The relation between the notions in the theory formalisation and those used in the text generator's application domain is also again discussed. Finally, we present two other theories of dialogue coherence which we have used in our system.

In both the global focus and local focus theories we present, the dialogue evolves when communications are output. The state of the dialogue therefore evolves from state to state each time a communication C is processed. Communications are used to exchange knowledge. Each communication has one *main subject*, i.e., the most important thing the communication is about, and possibly some other *subjects*, i.e., the other things the communication mentions, which serve as background information. The exact nature of the communications depends on the application domain and is discussed in chapter 4 in the case of our system.

Some details of the theories presented here, such as the rule ordering, cannot be completely justified during our formal presentation since they depend on how dialogues are perceived and not just on the mathematical correctness of the theories. They will be clarified when we present how the theories work in practice in chapters 5, 6 and 7. They will also be carefully evaluated in chapter 9.

3.2 Global focus theory

Our formalisation of a global focus theory is based on a simplification of Reichman's global focus theory (see section 2.3.5 and Lecœuche et al. (1999)). This theory is powerful enough to deal with a lot of dialogue phenomena while remaining simple enough to be implemented. The basic idea is that the dialogue is supported by a sequence of changes to a focus space set. The focus space set contains focus spaces which represent the information in focus during part of the dialogue. Each focus space contains all the things to which participants in a dialogue are attending at a certain point in a dialogue. If $\mathcal{S} = \{\mathcal{F}_1\}$ is the initial focus space set, only containing the initial focus space $\mathcal{F}_1$ at the beginning of the dialogue, then the set when t exchanges have been carried out in the dialogue is $\mathcal{S} = \{\mathcal{F}_1, \ldots, \mathcal{F}_n\}$ where each $\mathcal{F}_i$ contains some of the things spoken about in the dialogue. We may need to create several focus spaces ($\mathcal{F}_i$) since the dialogue may deal with more than one topic. Focus spaces have different activation levels (see section 2.3.5). We use three activation levels in our formalisation:

Active This is the space to which current communications are added. This space is unique at any given point in the dialogue. For example, at the beginning of the dialogue, focus space $\mathcal{F}_1$ is the active space.

Controlling These are the spaces expected to become active again when the space they control is closed. The controlling spaces form a tree: each space is controlled by at most one other space and a space cannot control a space already controlling it or controlling any of its controlling spaces. The controlling spaces serve one of two roles: they represent the point of view we have on things that are discussed in the active focus space (see section 2.2 and rule 3.3 below), or they simply represent things we have to return to after an interruption (see rule 3.6 below). These two functions are grouped together in the notion of controlling spaces because they give rise to the same dialogue structure, i.e., a return to the controlling space.

Closed These are the spaces which have been dealt with and are not expected to be returned to.

Communications cause things to be included in focus spaces. They can also cause focus space activations to be modified and/or focus spaces to be created. Because each communication can, at most, create one new focus space, the number of focus spaces, n, is always less or equal to the number of exchanges, t.

Our theory is composed of seven rules. The focus rules define the possible ways the dialogue may develop and the constraints to satisfy for choosing a particular way. Associated with an ordering, they represent what we are expecting to say next in the dialogue. Four rules create new spaces and two change the activation levels of existing spaces. The four creating moves are differentiated along two criteria: (1) moves that keep/do not keep the current focused entities in focus (2) moves that are expected/not expected to shift back to the current focus space once the newly created space is closed. The moves are based on four relations between the things that can be mentioned in the dialogue. Unlike ER relations, these relations are directional: for example, if a thing is in a direct relation with another there is not necessarily a direct relation from that other thing to the first. The four relations are:

Direct relation There is a direct relation from one thing, t_1, to another, t_2, if t_2 is closely related to t_1 and can be mentioned in the same focus space. (This means in particular that a thing is in direct relation with itself.) Conversely, if two things belong to the same focus space, they are considered in a direct relation.

Specialisation relation There is a specialisation relation from one thing, t_1, to another, t_2, if t_2 is more specific than t_1. In that case, the more specific thing can be discussed in the perspective of the more generic one (as was for example the star wars project in the light of US military strength in section 2.2).

Generalisation relation There is a generalisation relation from one thing, t_1, to another, t_2, if there is a specialisation relation in the other direction between t_2 and t_1, and no direct or specialisation relation links t_1 to t_2.

Simple relation There is a simple relation from one thing, t_1, to another, t_2, if t_2 is related to t_1 and no direct, specialisation or generalisation relation links t_1 to t_2.

The exact nature of these relations and, therefore, the meaning of words such as "closely" and "specific", depend on the application domain. In particular, the notion of specialisation considered here is not the same as the notion of specialisation found in many knowledge representation frameworks (usually in the form of "is-a" relations). For example, a specialisation relation may correspond to a part-of relation in one domain and a action-subaction relation in another. The issue of translating between domain relations and these relations is discussed in sections 3.2.2 and 4.4.1.

Because focus rules only indicate what changes in the focus spaces (most things stay the same), we are faced with a simple instance of the frame problem, i.e., how to indicate what changes and what does not (Kowalski 1979, ch. 6). We tackle this problem by taking into account the time for which a relation holds and adding information which indicates the end of this time period rather than deleting the relation itself (Kowalski and Sergot 1986). In particular, we consider the time in the dialogue as being the sequence of communications and associated moves that have been performed since the beginning of the dialogue. For example, at the start of the dialogue, the time is the empty sequence $\langle\rangle$ and after communications C_1, C_2 and C_3 have been output with their associated moves m_1, m_2 and m_3 respectively, the time is the sequence $\langle m_1(C_1), m_2(C_2), m_3(C_3)\rangle$. In other words, if the current time is t and the next communication output is C_u and the output is associated with move m, then the new time is $t \cdot \langle m(C_u)\rangle$ where $\cdot$ is the sequence concatenation operator. From a possible worlds perspective, the accessibility relation between worlds corresponding to this notion of time is defined as $W_1 \; R \; W_2 \Leftrightarrow time(W_1, t_1) \wedge time(W_2, t_2) \wedge t_2 = t_1 \cdot t$ where the *time* predicate associates a world with its time and t is a non empty sequence of moves. This relation is anti-reflexive $(\forall W. \neg(W \; R \; W))$, antisymmetric $(\forall W_1, W_2. W_1 \; R \; W_2 \rightarrow \neg(W_2 \; R \; W_1))$, transitive $(\forall W_1, W_2, W_3.(W_1 \; R \; W_2 \wedge W_2 \; R \; W_3) \rightarrow W_1 \; R \; W_3)$ and linear on the left $(\forall W_1, W_2, W_3.(W_1 \; R \; W_3 \wedge W_2 \; R \; W_3) \rightarrow (W_1 = W_2 \vee W_1 \; R \; W_2 \vee W_2 \; R \; W_1))$. From these properties we can deduce (Haton et al. 1991, p. 189) that each world has only one past (the dialogue history) and several possible futures (depending on which moves are available).

We define the $\succeq$ relation on time in the following way: $t_1 \succeq t_2 \Leftrightarrow \exists t_3.(t_1 = t_2 \cdot t_3)$. The relation $\succ$ is then defined as $t_1 \succ t_2 \Leftrightarrow t_1 \succeq t_2 \wedge \neg(t_1 = t_2)$. Relations that begin to hold at a certain time are written $start(P, t)$ where P is the relation and t the time. Relations that stop holding at a certain time are written $end(P, t)$ where P is again the relation and t the time. In order to know if a relation holds at a certain time, we define the predicate $hold(P, t)$. The definition of this predicate is $hold(P, t) \Leftrightarrow \exists t_{start}.(t \succeq t_{start} \wedge start(P, t_{start}) \wedge \neg\exists t_{end}.(t \succeq t_{end} \wedge t_{end} \succ t_{start} \wedge end(P, t_{end})))$.

Because the notation used above is quite cumbersome, we will use some shortcuts. In particular, we will write $hold(P(A_1, \ldots, A_n), t)$ as $P_t(A_1, \ldots, A_n)$. The same applies for infix operators. For example, inclusion in a set at time t is noted $\in_t$. We will also often ignore the exact nature of the time sequence, only considering how many communications have been output. The exact sequence can be re-constructed from the dialogue history so we do not lose any information in doing so. For example, the time at the start of the dialogue will be $t = 0$ and after three communications have been output $t = 3$. We will ignore the time altogether if it does not influence the truth value of the predicate.

The notation used to represent the rules used in the theory is shown in table 3.1. Some immediate properties of the theory are given in table 3.2.

In the following rules, the time variable, t, the communication variable, C, and all variables appearing on both sides of the implications are universally quantified at the outermost level. All other variables are assumed to be existentially quantified unless explicitly universally quantified. When a new space is created, it is assigned number $n + 1$ where n is the number of existing spaces. n can be deduced from the dialogue time since it is equal to the number of moves in the dialogue history opening new focus spaces plus one. The value of $n + 1$ is computed by the function $new\ index(t)$ where t is the dialogue time. The interpretation of rule 3.1 (no change) for example is then the following: "If the current active focus space is $\mathcal{F}_i$ and if we can select a communication C such that its main subject is in direct relation with an element of $\mathcal{F}_i$, then we can perform a no change focus move. This move has the effect of putting all the subjects of C in space $\mathcal{F}_i$."

Rule 3.1 (No change) *The focus space does not change and new information is added to it. This rule is used to speak about things closely related to the things in the current active focus space.*

Table 3.1 Notation used to formalise the global focus rules

Expression	Interpretation
$\mathcal{F}_i$	A focus space
$active(\mathcal{F}_i)$	Focus space $\mathcal{F}_i$ is active
$controlling(\mathcal{F}_i, \mathcal{F}_j)$	Focus space $\mathcal{F}_i$ is controlling focus space $\mathcal{F}_j$
$closed(\mathcal{F}_i)$	Focus space $\mathcal{F}_i$ is closed
$\mathcal{C}$	$\mathcal{C}$ is a communication, i.e., a message to be output. It makes the dialogue evolve from time t to $t+1$.
$subject(\mathcal{C}, X)$	X is a subject of communication $\mathcal{C}$
$mainsubject(\mathcal{C}, X)$	X is the main subject of communication $\mathcal{C}$. The communication main subject is a communication subject
$dir(D, X_1, X_2)$	X_1 is in direct relation D with X_2
$spe(S, X_1, X_2)$	X_1 is in specialisation relation S with X_2
$gen(G, X_1, X_2)$	X_1 is in generalisation relation G with X_2
$sim(R, X_1, X_2)$	X_1 is related to X_2 by simple relation R

Table 3.2 Immediate properties of global focus theory

$\forall t.(t \geq 0 \rightarrow \exists \mathcal{F}_i.(active_t(\mathcal{F}_i) \wedge \forall \mathcal{F}_j.(active_t(\mathcal{F}_j) \rightarrow \mathcal{F}_j = \mathcal{F}_i)))$	At any point in the dialogue there is one and only one active focus space
$\forall t, X_1.\exists D.dir_t(D, X_1, X_1)$	Any element is in direct relation with itself
$\forall t, X_1, X_2.((X_1 \in_t \mathcal{F}_i \ \wedge\ X_2 \in_t \mathcal{F}_i) \rightarrow \exists D.dir_t(D, X_1, X_2))$	If two elements are in the same focus space, they are in direct relation
$\forall t, X_1, X_2.(\exists S.spe_t(S, X_1, X_2) \rightarrow \exists G.gen_t(G, X_2, X_1))$	If there is a specialisation relation between X_1 and X_2, there is a generalisation relation between X_2 and X_1

$$\left(\begin{array}{r} active_t(\mathcal{F}_i) \wedge \\ X_1 \in_t \mathcal{F}_i \wedge \\ mainsubject(\mathcal{C}, X_2) \wedge \\ dir_t(D, X_1, X_2) \end{array}\right) \rightarrow \forall X.(subject(\mathcal{C}, X) \rightarrow start(X \in \mathcal{F}_i, t')$$

where t' stands for $t \cdot \langle no\,change(\mathcal{C})\rangle$.

Rule 3.2 (Resetting) *A new focus space is created. This rule is used to speak about more abstract things than the ones in the current active focus. It may therefore serve to give background information on the things in the current active focus.*

$$\left(\begin{array}{r} active_t(\mathcal{F}_i) \wedge \\ X_1 \in_t \mathcal{F}_i \wedge \\ mainsubject(\mathcal{C}, X_2) \wedge \\ gen_t(G, X_1, X_2) \end{array}\right) \rightarrow \left(\begin{array}{l} end(active(\mathcal{F}_i, t') \\ \wedge\ start(closed(\mathcal{F}_i), t') \\ \wedge\ start(active(\mathcal{F}_{new\ index(t')}), t') \\ \wedge\ \forall X.(subject(\mathcal{C}, X) \\ \quad \rightarrow start(X \in \mathcal{F}_{new\ index(t')}, t')) \end{array}\right)$$

where t' stands for $t \cdot \langle resetting(\mathcal{C})\rangle\rangle$.

Rule 3.3 (Additive) *A new focus space is created. It is controlled by the current active space. Entities in the current active space are copied to the new space. This rule is used to speak about things that are more precise than the ones in the current active focus space. The new things are discussed from the perspective of the current active focus space.*

$$\begin{pmatrix} active_t(\mathcal{F}_i) \wedge \\ X_1 \in_t \mathcal{F}_i \wedge \\ mainsubject(\mathcal{C}, X_2) \wedge \\ spe_t(S, X_1, X_2) \end{pmatrix} \rightarrow \begin{pmatrix} end(active(\mathcal{F}_i), t') \\ \wedge\ start(active(\mathcal{F}_{new\ index(t')}), t') \\ \wedge\ start(controlling(\mathcal{F}_i, \mathcal{F}_{new\ index(t')}), t') \\ \wedge\ \forall X.(X \in_t \mathcal{F}_i \rightarrow start(X \in \mathcal{F}_{new\ index(t')}, t')) \\ \wedge\ \forall X.(subject(\mathcal{C}, X) \\ \qquad \rightarrow start(X \in \mathcal{F}_{new\ index(t')}, t')) \end{pmatrix}$$

where t' stands for $t \cdot \langle additive(\mathcal{C}) \rangle$.

Rule 3.4 (Generating) *A new focus space is created. Entities in the current active space are copied to the new space. Any controlling relation is passed from the current active space to the newly created space since we may not expect to come back to the current active space but we still expect to come back to its potential controlling space. This rule is used to speak about things related to the things in the current active focus space but not closely associated with them.*

$$\begin{pmatrix} active_t(\mathcal{F}_i) \wedge \\ X_1 \in_t \mathcal{F}_i \wedge \\ mainsubject(\mathcal{C}, X_2) \wedge \\ sim_t(R, X_1, X_2) \end{pmatrix} \rightarrow \begin{pmatrix} end(active(\mathcal{F}_i), t') \\ \wedge\ start(closed(\mathcal{F}_i), t') \\ \wedge\ start(active(\mathcal{F}_{new\ index(t')}), t') \\ \wedge\ \forall X.(X \in_t \mathcal{F}_i \rightarrow start(X \in \mathcal{F}_{new\ index(t')}, t')) \\ \wedge\ \forall X.(subject(\mathcal{C}, X) \\ \qquad \rightarrow start(X \in \mathcal{F}_{new\ index(t')}, t')) \\ \wedge\ controlling_t(\mathcal{F}_j, \mathcal{F}_i) \\ \qquad \rightarrow start(controlling(\mathcal{F}_j, \mathcal{F}_{new\ index(t')}), t') \end{pmatrix}$$

where t' stands for $t \cdot \langle generating(\mathcal{C}) \rangle$.

Rule 3.5 (Pop) *A controlling space becomes active again. This rule is used to come back to a space that was expected to be reactivated. This rule is always used in conjunction with one of the other rules.*

$$\begin{pmatrix} active_t(\mathcal{F}_i) \wedge \\ controlling_t(\mathcal{F}_j, \mathcal{F}_i) \end{pmatrix} \rightarrow \begin{pmatrix} end(active(\mathcal{F}_i), t') \\ \wedge\ start(closed(\mathcal{F}_i), t') \\ \wedge\ start(active(\mathcal{F}_j), t') \end{pmatrix}$$

where t' stands for $t \cdot \langle pop(\mathcal{C}) \rangle$.

Rule 3.6 (Digressing) *A new focus space is created. It is controlled by the current active space. This rule is used to change the focus of the dialogue for a period of time after which the dialogue will resume where it was interrupted.*

$$\left(\begin{array}{r} active_t(\mathcal{F}_i) \wedge \\ mainsubject(\mathcal{C}, X_1) \wedge \\ \forall R.(\neg\exists X.(X \in_t \mathcal{F}_i \wedge \\ (dir_t(R,X,X_1) \vee spe_t(R,X,X_1) \\ \vee gen_t(R,X,X_1) \vee sim_t(R,X,X_1)))) \wedge \\ \forall j.(X_1 \notin_t \mathcal{F}_j) \end{array}\right) \rightarrow$$

$$\left(\begin{array}{l} end(active(\mathcal{F}_i), t') \\ \wedge\ start(active(\mathcal{F}_{new\ index(t')}), t') \\ \wedge\ start(controlling(\mathcal{F}_i, \mathcal{F}_{new\ index(t')}), t') \\ \wedge\ \forall X.(subject(\mathcal{C}, X) \\ \qquad \rightarrow start(X \in \mathcal{F}_{new\ index(t')}, t')) \end{array}\right)$$

where t' stands for $t \cdot \langle digressing(\mathcal{C})\rangle$.

Rule 3.7 (Reopening) *An old space becomes active again. This rule is used to come back to a topic that was considered dealt with. There are two ways a topic may be re-introduced in a dialogue: (1) we realise we forgot to say something about it and come back to it (2) the topic is discussed from another perspective (e.g., we could discuss the Star Wars project from a military or technological point of view - see section 2.2). The reopening move only addresses the first type of re-introduction. For the second type a new focus space would be created since the dialogue context (i.e., the controlling space) has changed.*

$$\left(\begin{array}{r} active_t(\mathcal{F}_i) \wedge \\ closed_t(\mathcal{F}_j) \wedge \\ mainsubject(\mathcal{C}, X_1) \wedge \\ X_1 \in_t \mathcal{F}_j \wedge \\ \forall R.(\neg\exists X.(X \in_t \mathcal{F}_i \wedge \\ (dir_t(R,X,X_1) \vee spe_t(R,X,X_1) \\ \vee gen_t(R,X,X_1) \vee sim_t(R,X,X_1)))) \end{array}\right) \rightarrow \left(\begin{array}{l} end(active(\mathcal{F}_i), t') \\ \wedge\ start(closed(\mathcal{F}_i), t') \\ \wedge\ end(closed(\mathcal{F}_j), t') \\ \wedge\ start(active(\mathcal{F}_j), t') \\ \wedge\ \forall X.(subject(\mathcal{C}, X) \\ \quad \rightarrow start(X \in \mathcal{F}_j, t')) \end{array}\right)$$

where t' stands for $t \cdot \langle reopening(\mathcal{C})\rangle$.

The dialogue starting state for this theory is $active_0(\mathcal{F}_1)$ with $\mathcal{F}_1 =_0 \{\}$.

Now that the rules have been formalised, it is possible to prove that some properties hold. For example, we can verify that in any dialogue, there is only one active space. This can be shown by induction on the dialogue time. At time $t = \langle\rangle$, the property is verified since there is only one focus space and this space is active. If we then suppose that for any time $t \prec T$, the property holds, it also holds at time $t = T$ because (1) every move starting a new active relation ends the current active relation, (2) no move ends the current active relation without starting a new one. Checking these properties is difficult in most other focus theories because they are not formalised.

Although the main characteristics of Reichman's approach are conserved, the theory presented above is a simplified version of the original approach. In particular, fewer types of focus spaces and fewer types of relations between focus spaces are used. Also, each focus space contains less information than in Reichman's theory. On the other hand, this theory is richer than some other theories since we distinguish between the normal evolution of the dialogue and digressions, and we also keep track of closed spaces. This balance between simplicity and complexity seems adequate for our purposes (see chapter 7).

3.2.1 Rule ordering

Several rules may have their preconditions satisfied at the same time. We prefer then to apply the rule that maintains the focus if possible, or minimises its movement. Changing focus, i.e., changing the entities to which participants are paying attention, does require some cognitive processing. Therefore, minimising focus movement reduces the cognitive overhead needed to understand the dialogue. We minimise the focus movements by presenting general concepts before specialised ones and by avoiding references to unrelated concepts. Rules are therefore applied in the following preference order:

> no change > resetting > additive > generating > pop > digressing (in initial active space) > reopening (in initial active space).

This means that we first try to find a communication to be output which would allow a no change move. Then, if such a communication does not exist, we would try to find a resetting communication and so on until a communication is found. When a pop move is done, moves are searched again in preference order in the reactived space. In other words, if we have not been able to find any no change, resetting, additive or generating move in the current active space and a controlling space exists, we pop to this space and start searching again for a no change, resetting, additive or generating move. Digressing and reopening moves are not considered during this search. Since these moves disrupt the flow of the dialogue, it seems better to disrupt the current active space rather than to move first to a more general space and disrupt it. Several pop moves can be made if the controlling space is itself controlled by other spaces. If the search for a move fails, then we come back to the initial active space and try to find a digressing or reopening move.

There is an exception to this ordering. If the main subject of a possible communication to output is already a member of a space controlling the current active space then a pop is the preferred move. This avoids reintroducing concepts that we are expecting to return to later in the dialogue.

3.2.2 Translation rules

Because the focus rules we presented in section 3.2 are based on generic concepts such as direct, specialisation or generalisation relations which are not necessarily used in the application domain of the system, we need to bridge the gap between the representation used in this domain and the representation used by the focus rules (Kittredge et al. 1991; Jönsson and Dahlbäck 1997). This is done by "translation rules". These rules also allow the use of this focus theory in different domains.

We have created a set of translation rules to map the specification model used by our elicitation system to the generic concepts used by our formalised focus theory. The rules are described in detail in chapter 4 where the specification model is also discussed.

3.3 Local focus theory

Our formalisation of a local focus theory is based on a simplification of the extended centering theory (see section 2.4.1). This theory has been extensively studied which makes it easier to formalise. Moreover, the various problems remaining with it are unlikely to create problems in the kind of dialogues we deal with. The basic idea is that each sentence introduces things into the dialogue. The following sentence then selects one of these things as the topic of the dialogue. For example, assuming I just said: "I bought a new hat at the shop round the corner yesterday", if the next sentence is: "I am going to a wedding" then the dialogue is about me but if the next sentence is: "It is nice, isn't it" then the dialogue is about the hat.

Each sentence in the dialogue is therefore associated with some "forward-looking centres", i.e., the things spoken about in the sentence, and to a "backward-looking centre", i.e., the topic selected by the sentence from the forward-looking centres of the previous sentence. Forward-looking centres are ranked. The most highly ranked centre is called the preferred centre and is expected to become the backward-looking centre of the next sentence, although this is of course not necessary. The ranking of the forward-looking centres depends on the importance of the things introduced.

When a new communication is output, the new backward-looking centre is the most highly ranked element of the current forward-looking centres realised in the new sentence. The new forward-looking centres are the things introduced by the new sentence.

The theory is composed of five rules. These rules and the ordering associated with them constrain how the dialogue may evolve. Focus rules only indicate what *changes* in the dialogue state. Everything else stay the same.

The notation used to represent the rules is shown in table 3.3. We use the same time framework as in the global focus theory formalisation (see section 3.2). Note that if both theories are used at the same time, the dialogue time becomes a sequence of tuples where each tuple is composed of a global focus move and a local focus move. This can be generalised for more than two focus theories. Some immediate properties of the local focus theory are given in table 3.4. In the following rules, the time variable, t, the communication variable, C, and all variables appearing on both sides of the implications are universally quantified at the outermost level. All other variables are assumed to be existentially quantified unless explicitly universally quantified. The interpretation of rule 3.8 (continuation) for example is then the following: "If the current backward-looking centre is B and if we can select a communication C such that its backward-looking centre with respect to the current dialogue state and its preferred centre are both equal to B then we can perform a continuation focus move. This move has the effect of changing the current forward-looking centres to those of C and the preferred centre to that of C."

Rule 3.8 (Continuation) *The new backward-looking centre is the same as for the previous sentence and is also the new preferred centre. This rule is used to continue to speak about the same thing.*

$$\left(\begin{array}{r} Cb_t(B) \wedge \\ Cp_t(P) \wedge \\ Cf_t(F) \wedge \\ Cb_t(C, NewB) \wedge \\ Cp_t(C, NewP) \wedge \\ Cf_t(C, NewF) \wedge \\ B = NewB = NewP \end{array}\right) \rightarrow \left(\begin{array}{l} end(Cp(P), t') \\ \wedge\, start(Cp(NewP), t') \\ \wedge\, end(Cf(F), t') \\ \wedge\, start(Cf(NewF), t') \end{array}\right)$$

where t' stands for $t \cdot \langle continuation(C) \rangle$.

Rule 3.9 (Retaining) *The new backward-looking centre is the same as for the previous sentence but it is different from the new preferred centre. This rule is used to introduce a*

Table 3.3 Notation used to formalise the local focus rules

Expression	Interpretation
$Cb(X)$	X is the backward-looking centre
$Cf(X_1, \ldots, X_n)$	$X_1, \cdots, X_n$ are the forward-looking centres
$Cp(X)$	X is the preferred centre
$\mathcal{C}$	$\mathcal{C}$ is a communication, i.e., a message to be output. It makes the dialogue evolve from time t to $t+1$.
$Cb(\mathcal{C}, X)$	X would be the new backward-looking centre if communication $\mathcal{C}$ was output
$Cf(\mathcal{C}, X_1, \ldots, X_n)$	$X_1, \cdots, X_n$ would be the new forward-looking centres if communication $\mathcal{C}$ was output
$Cp(\mathcal{C}, X)$	X would be the new preferred centre if communication $\mathcal{C}$ was output
$subject(\mathcal{C}, X)$	X is a subject of communication $\mathcal{C}$
$mainsubject(\mathcal{C}, X)$	X is the main subject of communication $\mathcal{C}$. The communication main subject is a communication subject

Table 3.4 Immediate properties of local focus theory

$\forall t, X, X_1, \ldots, X_n.(Cp_t(X) \land Cf_t(X_1, \ldots, X_n) \rightarrow X \in \{X_1, \ldots, X_n\})$	The preferred centre is always one of the forward-looking centres

new topic in the dialogue with respect to the current one.

$$\begin{pmatrix} Cb_t(B) \land \\ Cp_t(P) \land \\ Cf_t(F) \land \\ Cb_t(\mathcal{C}, NewB) \land \\ Cp_t(\mathcal{C}, NewP) \land \\ Cf_t(\mathcal{C}, NewF) \land \\ B = NewB \land \\ \neg NewB = NewP \end{pmatrix} \rightarrow \begin{pmatrix} end(Cp(P), t') \\ \land\ start(Cp(NewP), t') \\ \land\ end(Cf(F), t') \\ \land\ start(Cf(NewF), t') \end{pmatrix}$$

where t' stands for $t \cdot \langle retaining(\mathcal{C})\rangle$.

Rule 3.10 (Smooth-shift) *The new backward-looking centre is not the same as for the previous sentence but is the same as the new preferred centre. This rule is used to shift the focus of the dialogue to a new topic which is supposed to be discussed for some time.*

$$\begin{pmatrix} Cb_t(B) \wedge \\ Cp_t(P) \wedge \\ Cf_t(F) \wedge \\ Cb_t(\mathcal{C}, NewB) \wedge \\ Cp_t(\mathcal{C}, NewP) \wedge \\ Cf_t(\mathcal{C}, NewF) \wedge \\ \neg B = NewB \wedge \\ NewB = NewP \end{pmatrix} \rightarrow \begin{pmatrix} end(Cb(B), t') \\ \wedge\ start(Cb(NewB), t') \\ \wedge\ end(Cp(P), t') \\ \wedge\ start(Cp(NewP), t') \\ \wedge\ end(Cf(F), t') \\ \wedge\ start(Cf(NewF), t') \end{pmatrix}$$

where t' stands for $t \cdot \langle smooth\,shift(\mathcal{C})\rangle$.

Rule 3.11 (Rough-shift) *The new backward-looking centre is not the same as for the previous sentence and is different from the new preferred centre. This rule is used to shift the focus of the dialogue to some new topic.*

$$\begin{pmatrix} Cb_t(B) \wedge \\ Cp_t(P) \wedge \\ Cf_t(F) \wedge \\ Cb_t(\mathcal{C}, NewB) \wedge \\ Cp_t(\mathcal{C}, NewP) \wedge \\ Cf_t(\mathcal{C}, NewF) \wedge \\ \neg B = NewB \wedge \\ \neg NewB = NewP \end{pmatrix} \rightarrow \begin{pmatrix} end(Cb(B), t') \\ \wedge\ start(Cb(NewB), t') \\ \wedge\ end(Cp(P), t') \\ \wedge\ start(Cp(NewP), t') \\ \wedge\ end(Cf(F), t') \\ \wedge\ start(Cf(NewF), t') \end{pmatrix}$$

where t' stands for $t \cdot \langle rough\,shift(\mathcal{C})\rangle$.

Rule 3.12 (No centre) *The backward-looking centre cannot be computed. It is considered by-default to be the same as the preferred centre. This rule is used to determine the by-default focus of the dialogue when none can be computed otherwise.*

$$\begin{pmatrix} Cb_t(B) \\ Cp_t(P) \\ Cf_t(F) \\ Cb_t(\mathcal{C}, \emptyset) \wedge \\ Cp_t(\mathcal{C}, NewP) \wedge \\ Cf_t(\mathcal{C}, NewF) \end{pmatrix} \rightarrow \begin{pmatrix} end(Cb(B), t') \\ \wedge\ start(Cb(NewP), t') \\ \wedge\ end(Cp(P), t') \\ \wedge\ start(Cp(NewP), t') \\ \wedge\ end(Cf(F), t') \\ \wedge\ start(Cf(NewF), t') \end{pmatrix}$$

where t' stands for $t \cdot \langle no\,centre(\mathcal{C})\rangle$.

The dialogue starting state for this theory is $Cb_0(\emptyset)$, $Cp_0(\emptyset)$, $Cf_0(\emptyset)$.

Most of these rules are a formalisation of the centering theory adapted for generation. The main difference compared to the original centering theory is the explicit mention of communications in rules. For example, the continuation rule is simply expressed as $Cb_t(B) \wedge Cb_{t+1}(B) \wedge Cp_{t+1}(B)$ in the centering theory. However, it is unclear when we should consider that t becomes $t+1$. Most of the text analyses based on centering are made at the sentence level, i.e., t increases with each sentence, but some are made at the phrase level. By adding communications in the rules, we explicitly associate the increment in t with the output of a communication. In our system, each communication is then associated with a sentence. However, this could be changed and communications could be associated with phrases if need be. Linking the communications and the focus rules also enables us to make the link between the centres in the text and the communication structures (see section 3.3.2).

The no centre rule is new. This rule is necessary for text generation. In the case of text understanding, a sentence without backward-looking centre, such as the first sentence of a text, just sets things up but does not trigger any rules. This is why an anaphora in the first sentence of a text cannot be resolved by the centering theory since no rule can yet apply. However, for text generation, we need to trigger a rule to produce sentences, even when the focus centres cannot be computed. This is the role of the no centre rule.

3.3.1 Rule ordering

Several rules may have their preconditions satisfied at the same time. We prefer then to apply the rule which allows a smooth introduction of new topics in the dialogue. Rules are therefore applied in the following preference order:

continuation > smooth-shift > retaining > rough-shift > no centre.

This is a variation on the original ordering where retaining is preferred to smooth-shift (see section 2.4.1). This means that we first try to find a communication to be output which would allow a "continuation" move. Then, if such a communication does not exist, we would try to find a "smooth-shift" communication and so on until a communication is found. Preferring smooth-shift to retaining introduces new things in the dialogue as soon as they can be. This is equivalent to McKeown's rules (see section 2.6.1) and is a quite general principle for text generation. While text understanding is by-default conservative, i.e., there is no reason to prefer a shift move over continuation and retaining moves without some cues in the dialogue (Allen 1987, p. 401), text generation cannot adopt the same strategy. There is a need to actively push new topics in the dialogue when needed. This is why the ordering of the rules is modified. However, since the hearer of the dialogue will still prefer the initial ordering of the rules as defined in the centering theory, it is important to signal explicitly what kind of move is being done in the dialogue. (The idea of marking transition in dialogue is also used in Mittal et al. (1998).) This is done during the text generation process by using the right sentence structure (see section 5.4.1) and the right pronouns (see section 5.4.2) to indicate the moves made.

3.3.2 Translation rules

Because the focus rules we presented in section 3.3 are based on generic concepts such as backward or forward looking centres which are not necessarily used in the application domain, we need again to bridge the gap between these representations. However, in the case of the theory presented above, the translation rules are particularly simple since they only depend on the structure of the sentences and not on their semantics:

- The forward-looking centres correspond to everything that is mentioned in the sentence:
  $\forall t, \mathcal{C}, X, X_1, \ldots, X_n. (Cf_t(\mathcal{C}, X_1, \ldots, X_n) \wedge X \in X_1, \ldots, X_n \Leftrightarrow \mathit{subject}(\mathcal{C}, X))$.

- The preferred centre corresponds to the main subject of the sentence: $\forall t, C, X.(Cp_t(C,X) \Leftrightarrow mainsubject(C,X))$. The other forward-looking centres are not particularly ordered as they only present background information. Note that the selection of the preferred centre is not the same as the by-default selection of a topic in McKeown's system (see section 2.6.1). In that system, the knowledge pool is flat, i.e., all pieces of information have the same importance, and the system is in charge of organising them. It uses a by-default assignment of topicality when no specific rules can apply. In our system, the importance level, i.e., the main subject, is defined by the elicitation module (see section 4.3). This affects how the communication is output, rather than the other way round. If we wanted to let the dialogue manager decide on the relative importance of the pieces of information, we could ignore the elicitation module's suggestion and just specify that the main subject is one of the subjects. The dialogue manager would then select as main subject a subject that allows the best focus rule possible to apply.

- The backward-looking centre of the sentence depends on the previous sentences (see section 2.4.1). It is, in order of preference:

  1. The preferred centre of the previous sentence if this centre is realised in the current sentence.
  2. One of the elements of the previous sentence's forward-looking centres which is realised in the current sentence. The exact element will be the one making the best local focus move possible (see section 3.3.1).
  3. The empty set if the previous cases do not apply. In this case, it will be considered as the same as the preferred centre of the current sentence by the no centre rule (see section 3.3).

3.4 Other theories of dialogue coherence

Managing focus is only one of the tasks that a dialogue manager must perform. In this section we present other theories which tackle some other issues. These theories are used in addition to the two focus theories presented above.

3.4.1 Present-first theory

The focus theories do not distinguish between communications with different "speech acts" (Austin 1962) such as a communication presenting something or a communication asking something. As far as the coherence of the dialogue is concerned, these two communications are equivalent if they are about the same thing. Obviously, this may lead to some problems especially during an elicitation dialogue. For example, asking questions about things that have not yet been presented would make the dialogue very difficult to understand. On the other hand, presenting things that have not yet been discussed is usually perfectly acceptable. In order to deal with this situation, we have written a simple theory. This theory is not a focus theory since it does not take into account what users are paying attention to in the dialogue. It simply gives a preference to communications informing users of something over communications asking users for something. This can be summarised by the rule:

Rule 3.13 (Present first) *Prefer presentation communications over question communications*

Translation rules

There is again a need to bridge the gap between the representation used for communications in the application domain and the representation used by the present-first theory which distinguishes between presentation communications and question communications. This is done by rules depending on the types of communications that the system using this theory produces. In the case of our system, these rules are presented in section 4.3.1 where the different types of communications used in our system are also discussed.

3.4.2 Other focus theories

Although not studied in detail, some other focus theories have been implemented in our system. In particular, we implemented a simplified version of Walker's theory. The main difference from the global theory presented above is that the focus spaces are replaced by a single space acting like a First-In-First-Out (FIFO) data store. This is a rough approximation of the memory mechanism of the theory. Since new things put in focus may displace old things, it may become necessary to re-introduce things in the dialogue. This is the role of Information Redundant Units (IRU). When this becomes necessary, the theory predicts an IRU move which associates a presentation communication, re-introducing an element in focus, with the current sentence to output. This presentation communication is then output before the main communication (see section 5.4.3). There are some problems with this theory though. For example, it is not clear if the focus space should behave like a simple FIFO data store or should give priority to some elements, thus trying to keep them in memory. It seems likely that important elements in the dialogue are more easily remembered than others. However, finding a prioritisation function depends on expectations of what the dialogue will be about. This is still an open problem (see section 11.2.2).

Chapter 4

System Architecture

Objectives

The objectives of this chapter are to:

- Present a high-level description of our system,
- Describe the elicitation module and especially its representation framework and its basic actions,
- Describe the dialogue manager and especially its relation with the elicitation system,
- Formalise the translation rules between the elicitation module's representation framework and the focus rules representation.

4.1 Introduction

As presented in chapter 1, our elicitation system is based on a general architecture composed of two main components: an elicitation module and a dialogue manager. The elicitation module manages the specification. In particular, it creates entities and relations that it deems necessary depending on its domain knowledge and the user's utterances. It also generates and interprets communications. The dialogue manager manages the dialogue with users. In particular, it ranks communications in the communication pool and select one of them to be presented to users. It also interprets user's answers. The elicitation module and the dialogue manager may have their own sources of knowledge. A buffer called the *communication pool* serves as a means of communication between these two processes. This is shown in the data flow diagram presented in figure 4.1. A data flow diagram presents how data and control information flow in the system. Data items are entered in the system by external elements. They are transfered by data flows and are transformed by data processing elements. Finally, they can be stored in data stores which have an infinite storage capacity or buffers which have finite capacity. Data items put in a buffer are removed when read to free up space for other items. Transfer and processing of data are considered instantaneous. The graphical notation used in this figure is explained in table 4.1 and is discussed in detail in Ward (1986). As its name indicates, the communication pool contains communications.

These can be thought of as messages exchanged between the elicitation module and the dialogue manager. They contain information needed to produce and interpret the dialogue. This information consists of the speech act to perform, e.g., present and ask, the domain knowledge communicated, and possible further information about the communication itself, e.g., which part of the communication is important.

Figure 4.1 General architecture

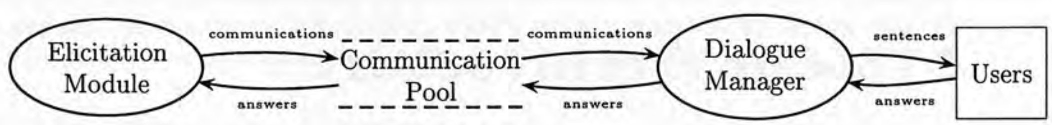


Table 4.1 Data Flow Diagram notation

Graphical notation	Meaning	Graphical notation	Meaning
External	External Element	→	Discrete Data Flow
		Buffer	Buffer
Data	Data Processing	Data Store	Data Store

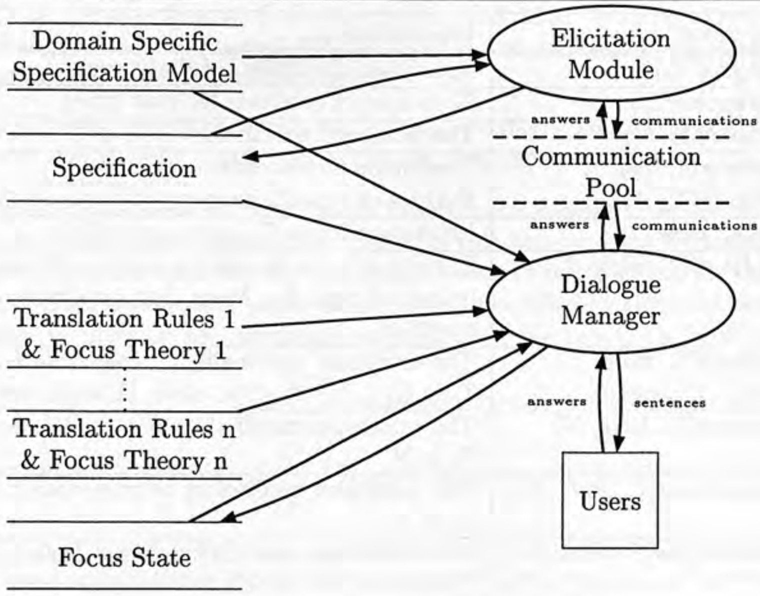
Our system is a special case of this general architecture. The main differences between our system and the general architecture are that:

- The dialogue manager and the elicitation module share the same domain knowledge. This makes the collaboration between these two processes easier (see section 4.4).

- In addition to the speech act to perform and the domain knowledge communicated, the communications only contain information about which part of the knowledge exchanged plays a prominent role, i.e., which part of the communication is its main subject (see sections 3.2 and 3.3). This information is sometimes mandatory anyway since we need to know exactly what the communication is about, e.g., when asking questions. The communications do not contain any other information on themselves or other communications. This makes the distinction in the roles played by the elicitation module, which reasons on the specification, and the dialogue manager, which organises the dialogue, clearer.

The resulting high-level data flow diagram of our system is given in figure 4.2. We describe it in more detail in the rest of this chapter. We also present how it relates to the formal theories presented in section 3. We conclude the chapter with an algorithm presenting how our system operates.

Figure 4.2 Elicitation system architecture



4.2 Elicitation module

The elicitation module is composed of two data stores and one transformation process (see figure 4.2). The two data stores are described below:

The **Domain Specific Specification Model** contains the domain specific knowledge about what can be elicited. This model is used during the elicitation process to direct the system. It is also used for correctness and completeness checking. For example, when our system works on WWW site design for research groups (which is its main application domain; see section 4.2.1 and chapter 7), this model contains knowledge about WWW sites, such as that they are composed of pages linked together.

The **Specification** represents the requirements that have been elicited. For example, when our system works on WWW site design for research groups, this data store could contain information such as a page, e.g., page 1, is linked to another one, e.g., page 2. The specification can also contain some "hypothetical" entities and relations that the elicitation system assumes will be created but have not yet been asserted by users. Hypothetical elements are used by the system to reason about the specification but are liable to be removed if users deny their existence. On the other hand, the existence of such elements can be confirmed by users, in which case their hypothetical status is removed.

These two data stores are based on a representation framework presented in section 4.2.1. The transformation process modifies the specification to take into account user's requirements while maintaining the specification integrity with respect to the domain specific specification model. The process also outputs communications to the communication pool to

Table 4.2 Entity-Relationship Model notation (model level)

Expression	Interpretation
$entityset(T_e)$	T_e is a set of entities (of that type)
$relationset(T_r)$	T_r is a set of relations (of that type)
$cardmin(T_e, N)$	The minimum cardinality of T_e is N
$cardmax(T_e, N)$	The maximum cardinality of T_e is N
$attribute(T_e, A, T_a)$	Entities of type T_e have an attribute A of type T_a
$role(T_r, Role)$	Relations of type T_r have a role *Role*
$rolefiller(T_r, Role, T_e)$	Entities of type T_e can be used to fill the role *Role* of relations of type T_r. More than one type of rolefiller can be specified
$cardmin(T_r, Role, N)$	The minimum cardinality of role *Role* for relation of type T_r is N
$cardmax(T_r, Role, N)$	The maximum cardinality of role *Role* for relation of type T_r is N
$rolecardmin(T_r, Role, N)$	The minimum cardinality of role *Role* for relation of type T_r is N
$rolecardmax(T_r, Role, N)$	The maximum cardinality of role *Role* for relation of type T_r is N

guide users during the requirements elicitation. This process is presented in section 4.2.2.

4.2.1 Representation framework

Table 4.3 Entity-Relationship Model notation (instance level)

Expression	Interpretation
$entity(E, T_e)$	Entity E is of type T_e
$attributevalue(E, A, V)$	Entity E has an attribute A of value V
$relation(R, T_r)$	Relation R is of type T_r
$rolefillervalue(R, Role, E)$	Entity E fills the role *Role* of relation R

In our system, the specification and specification model are based on an Entity-Relationship (ER) model (Chen 1976). An ER model consists of entity-sets which define the types of things we can speak about, relationship-sets which define the types of relations which can link entities together, and attributes which define the types that values describing entities or relationships can have. A model can then be instantiated by creating entities (of a given entity-set), by creating relationships (of a given relationship-set) linking them and by giving values (of a given type) to the attributes describing entities or relationships. The notation used to describe the model is presented in table 4.2. An ER model is not always the best nor the most complete way of describing requirements. In particular, dynamic behaviour is difficult to represent with an ER model. Other aspects of the specification such as the requirements rationale, sources and dependencies are not represented either. Even for static domains, richer representations exist, such as those of the KL-ONE family. However, we chose an ER model for its simplicity and because it is well suited for describing static domains such as the ones that we deal with (Davis 1988, 1993; Wieringa 1996).

We have augmented the original ER model with two additional types of constraints. These two additional types of constraints are:

- Cardinality restrictions on entity-sets can be specified. This enables us to define the minimum and maximum numbers of entities of a particular type.
- Cardinality restrictions on relationship roles can be specified. A role is the function played by entities in a relationship. When a role can be filled by entities of several different types, the minimum and maximum numbers of role-fillers for the role can be specified. This is especially useful when roles can be filled by either of two entities but not both (in this case the maximum cardinality for the role is 1). In all cases:

$$rolecardmin(T_r, Role, N) \geq \sum_{T_e} 1_{\cdot rolefiller(T_r, Role, T_e)} \times cardmin(T_r, Role, N)$$

$$rolecardmax(T_r, Role, N) \leq \sum_{T_e} 1_{\cdot rolefiller(T_r, Role, T_e)} \times cardmax(T_r, Role, N)$$

where $1_{\cdot p(x_1, .., x_n)} = 1$ if $p(x_1, \ldots, x_n)$ is true and 0 otherwise. These equations mean that the role cardinalities should impose more stringent constraints on the model than the constraints already imposed by the relation cardinality restriction. The equalities hold if not specified otherwise.

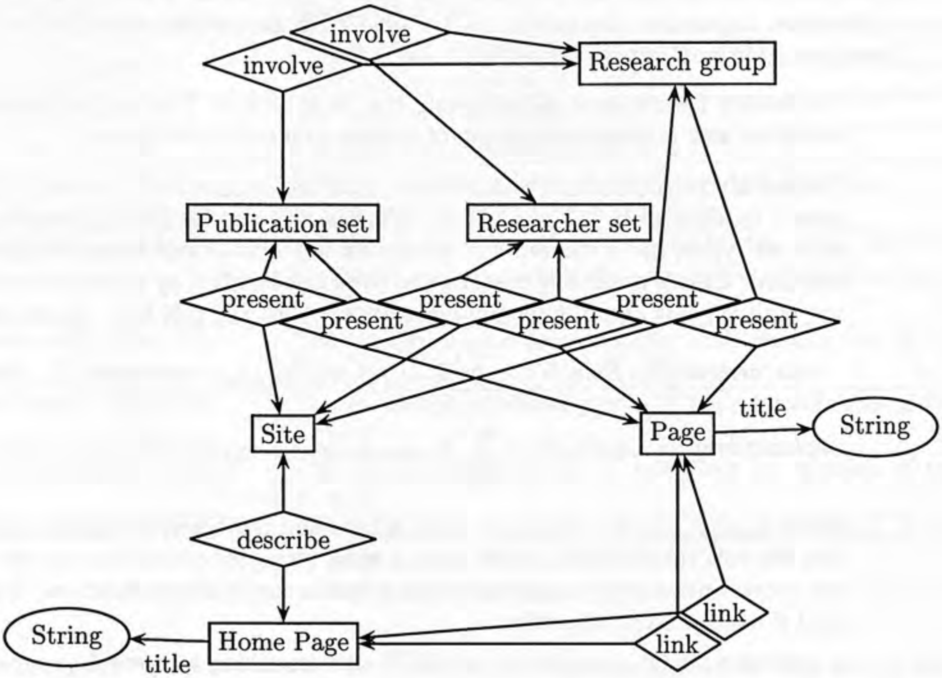
A part of a model constraining a WWW site describing a research group is presented in figure 4.3 and is detailed in appendix A.1. (The role names and the cardinality restrictions have been omitted from the figure.) The graphical notation used in the figure is described in table 4.4 and is discussed in detail in Wieringa (1996). This model shows that a research group is composed of a set of researchers and a set of publications. The research group and each of the sets can be presented by a site or a page. A site is itself presented by a home page. Home pages and pages can be linked together. We also use a shorter graphical notation presented in figure 4.4. (The cardinality restrictions have been omitted from the figure.) In this version, the relation `involve` has two roles, namely, `involved in` and `involving`. The latter can be filled by either `publication set` entities or `researcher set` entities. This is a short-cut for having two relations, one between the research group and the researcher set and one between the research group and the publication set. Since a site or a page can only present one entity among `research group`, `researcher set` and `publication set`, the `presenting` role has a maximum cardinality constraint of 1 (this does not appear in the figure).

The requirements are represented by instances of this ER model (see table 4.3). For example, the fact that a research group is presented by a site would be represented as in table 4.5. Hypothetical entities and relations in the specification are marked by question marks. (This is not shown in table 4.5.)

4.2.2 Elicitation module basic actions

The actions of the elicitation module are based on a limited set of basic actions. These basic actions are presented in table 4.6. They enable the module to create hypothetical things

Figure 4.3 Entity Relationship model



in the specification, to assert them, i.e., to make them definite, and to remove them from the specification. They also allow the module to add communications to and remove communications from the communication pool. Finally, some basic actions enable the module to ask for the value of role-filler and attributes. Using the basic actions, the elicitation module generates communications about changes that can be made in the specification and interprets user's answers depending on the current state of the specification and on the domain specific specification model. Depending on the user's answers, it modifies the specification. It can also remove communications from the communication pool. This gives much flexibility to the elicitation system in its communication decisions, but makes the task of the dialogue manager more complex since the content of the communication pool is less predictable than when communications can only be added to it. The elicitation system is

Table 4.4 Entity Relationship model notation

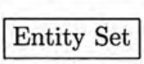
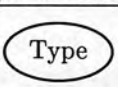
Graphical notation	Meaning	Graphical notation	Meaning
	Entity Set		Attribute type
	Relationship Set		Relationship role or Attribute

Figure 4.4 Short Entity Relationship model

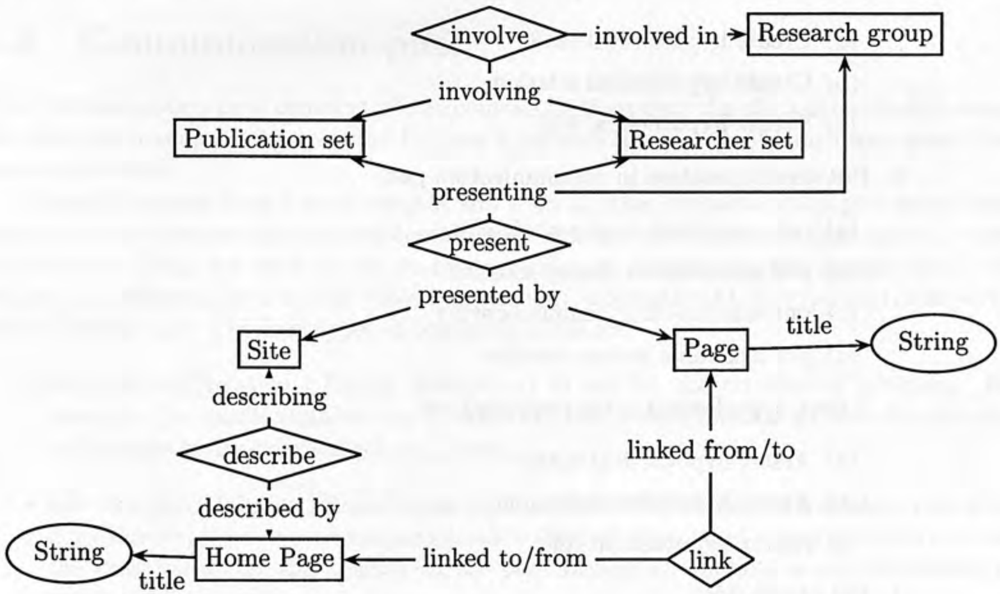


Table 4.5 Specification example

Specification
entity(rg1, research group)
entity(s1, site)
relation(p1, present)
rolefillervalue(p1, presenting, rg1)
rolefillervalue(p1, presented by, s1)

monotonic since elements that have been asserted in the specification cannot be removed. This could be changed by adding new basic actions. For example, a basic action could be created to remove existing relations and another to remove entities that are not linked by any relations.

In our system, the elicitation system outputs all the questions it can and makes all the modifications possible given the user's answers. The general algorithm driving the module is the following:

- Create all entities and relations that should exist. This depends on the minimal cardinalities specified in the domain specific specification model.
- Add *presentation communications* to the communication pool informing users of what has changed in the specification (see section 4.3). In particular, all entity and relation creations are presented as well as relations' new role fillers.
- Add *question communications* to the communication pool asking for missing informa-

Table 4.6 Elicitation basic actions

1. Create hypothetical entity/relation/role
   (a) Create hypothetical entity
   (b) Create hypothetical relation
   (c) Create hypothetical role
2. Put communication in communication pool
   (a) put presentation communication
   (b) put ask-creation communication
   (c) put ask-role-filler communication
   (d) put ask-value communication
3. Assert hypothetical entity/relation/role
   (a) Assert hypothetical entity
   (b) Assert hypothetical relation
   (c) Assert hypothetical role
4. Fill placeholder
   (a) Fill a role's role-filler
   (b) Fill an entity or relation attribute
5. Remove hypothetical entity/relation/role
   (a) Remove hypothetical entity
   (b) Remove hypothetical relation
   (c) Remove hypothetical role
6. Remove question from communication pool

tion, such as the creation of new entities or relations, and role-fillers (see section 4.3).

- Interpret answers to question communications and modify the specification accordingly.
- Remove question communications from the communication pool that would lead to a violation of constraints. This depends on the maximal cardinalities specified in the domain specific specification model and on the modifications made to the specification since the communication has been placed in the communication pool. For example, once it has been decided that a research group is to be presented by a page, it cannot be presented by a site. If a question communication about this possibility is present in the communication pool, it will be removed.

These actions are described in more detail in chapter 5 for communication generation actions and chapter 6 for communication interpretation actions.

4.3 Communication pool

The communication pool contains all the communications that the elicitation module wants the dialogue manager to output and the user's answers to these communications when they become available.

Communications have a main subject and a set of other subjects which give some background information for the communication (see section 3.1). There are four types of communication. They are used to ask or present different elements in the specification. We denote a communication in the following manner: `communication-type(mainsubject | other subjects)`. The four types of communication are:

- ask-creation(Relation | Entity, Entity-Set) to ask for the creation of relations. For example, the question about the creation of presentation relation **p1** between site #s1 and a page is ask-creation(p1 | s1, page).

- ask-role-filler(Relation | Role, Entity-Set) to ask for the identity of a relation role filler. For example, if we assume the presentation relation mentioned above has been created, the question about the identity of the page presenting the site is ask-role-filler(p1 | presented by, page).

- ask-value(Entity | Att) or ask-value(Relation | Att) to ask for the value of an attribute. For example, if we assume that the identity of the page discussed in the examples above is #pa1, its title would be obtained by communication ask-value(pa1 | title).

- present(Entity) or present(Relation | Rolefillers) to inform users of the existence of some entities or relations. For example, the creation of relation #p1 could be confirmed by communication present(p1 | s1, pa1).

The three "ask-" communications are used to question users about carrying out changes in the specification such as the creation of entities or relations. The "present-" communication is used to inform users of changes in the specification. For example, if the elicitation module wants to inform users of the existence of relation **p1** in the specification presented in table 4.5, it would do it by putting a presentation communication in the communication pool with basic action 2a. The main subject of the communication is the relation to be presented and the background information is composed of the two role-fillers of the relation. This is presented in table 4.7. The communication would be output in due time as a sentence like "Research group #rg1 is presented by site #s1". The way these communications are

Table 4.7 Communication pool example

Specification	Communication pool
entity(rg1, research group)	present(p1 | rg1, s1)
entity(s1, site)	
relation(p1, present)	
rolefillervalue(p1, presenting, rg1)	
rolefillervalue(p1, presented by, s1)	

created and output is described in detail in chapters 5 and 6.

The communication pool also contains any user's answers to the question communications. These are represented as `answer(communication, answer)`. The answer is either `yes` or `no` when answering yes/no questions, `new` or the identifier of an entity when answering questions about the identity of an element, or free text when giving the value of an attribute. Users can also redirect the dialogue (see section 6.4.3). Other types of interaction, such as asking for clarification, are not allowed.

4.3.1 Relation with the present-first theory

The different types of communication used in our system are related to different speech acts. Depending on the act performed, the communications are treated differently by the present-first theory (see section 3.4.1). The relation between communication types and speech acts is straightforward and is summarised in table 4.8. Presentation communications will therefore

Table 4.8 Communication types and speech acts

Communication type	Speech act
ask-creation	question[a]
ask-role-filler	question
ask-value	question
present	presentation

[a]In terms of Searle's classification of speech acts (Searle 1979, ch. 3), questions are "directives" (requesting answers from users) and presentations are "assertives" (presenting a partial state of the specification).

be preferred by the present-first theory over the various "ask-" communications.

4.4 Dialogue manager

The dialogue manager is composed of several data stores and one transformation process (see figure 4.2). The data stores are:

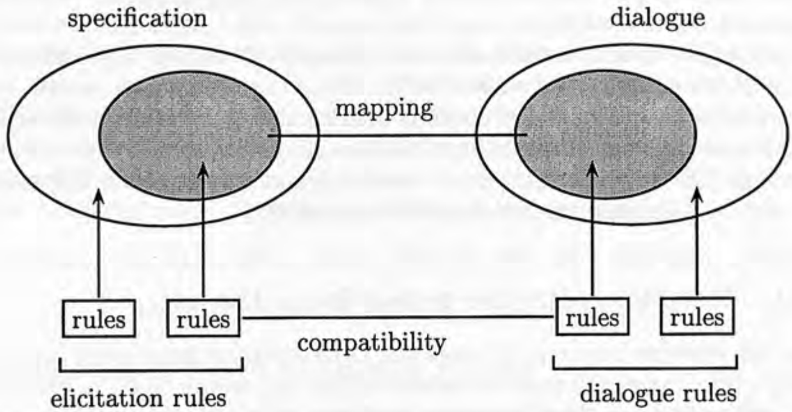
The **Translation Rules and Focus Theory** stores. Each contains the focus rules of a particular focus theory, their ordering and their associated translation rules as described in chapter 3 and section 4.4.1.

The **Focus State** which contains the current state of the dialogue.

The transformation process organises the dialogue by taking into account the communications which are in the communication pool and the focus rules. It selects a communication from the communication pool which maximises the coherence of the dialogue, generates a sentence which is presented to users and then interprets their answer and passes it to the elicitation module via the communication pool. Ideally, the dialogue manager would do this independently from the elicitation module, since we want to keep them as separate as possible in case we want to change the elicitation module for example. However, there is a clear problem with this: how would the dialogue manager know for example that #p1 is a presentation relation between research group #rg1 and site #s1 (see table 4.7) which is needed to output text?

A first solution is to force the dialogue manager and elicitation module to agree on what is created. In this case, a presentation communication is not a simple request from the elicitation module to inform users of a change in the specification but a request to create an equivalent element in the dialogue and to present this element to users. This request can be accepted or refused by the dialogue manager, depending on the state of the dialogue. The same applies to question communications with the added possibility of users influencing the creation of entities. The elicitation module and the dialogue managers are still free to work on their separate representations if this work does not appear in the communication pool. When the work influences what is in the communication pool, the rules making a change in the specification should be compatible with rules changing the dialogue representation (and vice-versa) and a mapping should ensure that the representations are kept synchronised. This approach is shown in figure 4.5. The gray areas in the figure represent the part of the representations which are kept synchronised. Although this approach is viable, it presents

Figure 4.5 Synchronising the elicitation module and dialogue manager representations (adapted from Lecœuche et al. (1998b))



some problems:

- Because the specification is closely linked to the dialogue manager representation, it is difficult to manage hypothetical elements (see section 4.2). It is not clear how the hypothetical elements in the specification should be related to the dialogue manager representation since the notion of hypothetical element does not exist in the dialogue, i.e., either we speak about something or we don't. If we create elements in the dialogue to represent hypothetical elements, then these elements may later refer to elements with another identity (when hypothetical elements are asserted in the specification), or to nothing (when hypothetical elements are removed from the specification). This problem requires careful thought about mapping the two representations. It could also be possible to solve this problem by separating more clearly the dialogue manager representation of entities spoken about in the dialogue from the specification's content (Luperfoy 1992). While this extra freedom may help handle hypothetical elements, a complex mapping process may still be required.

- The dialogue manager may require information which is not present in the communication pool. For example, the sentence "Research group #rg1 is presented by site #s1" (see section 4.3) could be output as "*The* research group is presented by site #s1" if we were sure, for example, that there can only be one research group in the specification. This information is not present in the communication pool.

A second approach which solves some of these problems is to allow the dialogue manager to have a direct access to the specification and specification model. The dialogue manager can only read these data stores which are managed by the elicitation module. The dialogue manager then has access to all the information it needs and the compatibility and mapping issues disappear. The dialogue manager may also manage its own data store for internal purposes. Of course, there is still a need for translation rules to transform the elicitation module representation into something useful for the dialogue manager and the focus theories it uses (see section 3.2.2). However, this problem is easier than the compatibility and mapping one because it is only one way, from specification and specification model to dialogue representation. Moreover, having a direct link between the specification model representation and natural language provides some other benefits too. Because natural language imposes some constraints on how concepts may relate to each other, it can be used as an organisation principle of the specification model representation (Bateman 1990, 1991; Burg and van de Riet 1995; Swartout and Smoliar 1987). This may lead to a re-work, and hopefully improvement, of the ER model if conflicts are detected with the natural language constraints. The natural language generation process can also serve as an "elementary sentence check" (Wieringa 1996, p. 182). Sentences describing small parts of the ER model are generated and read by a domain expert to check their validity.

4.4.1 Relation with the global focus theory

Since the dialogue manager accesses the representation framework used by the elicitation module, this framework must be related to the framework used by the global focus theory (see section 3.2) for them to operate. This is done by means of translation rules (see section 3.2.2). The translation rules are an engineering trick which enables us to re-use the knowledge already present in the domain specific specification model. This reduces the amount of work needed to build our system.

We present here a set of such rules. Other rules could be written to perform the translation. The main idea behind the rules presented here is to allow things that are related to the current focus and cannot be accessed easily otherwise to be mentioned immediately. (Their actual selection will depend on the current focus state and the focus rules.) These rules can be divided into two main categories: rules grouping elements together into a focus space, i.e., rules creating direct relations between elements, and rules defining the relations between focus spaces, i.e., rules creating specialisation or generalisation relations. These rules complete the general properties of the focus spaces (see section 3.2) with domain specific ones. In the following rules, all variables are assumed to be universally quantified unless explicitly existentially quantified. There are three rules for creating direct relations:

Rule D1 If two entities are in the same focus space, any relation (R) linking them is in a direct relation with them.

$$\begin{pmatrix} rolefillervalue_t(R, Role_1, E_1) \wedge \\ rolefillervalue_t(R, Role_2, E_2) \wedge \\ Role_1 \neq Role_2 \wedge \\ E_1 \in \mathcal{F}_{i,t} \wedge \\ E_2 \in \mathcal{F}_{i,t} \end{pmatrix} \rightarrow \begin{pmatrix} \exists D.dir_t(D, E_1, R) \\ \wedge\, \exists D'.dir_t(D', E_2, R) \end{pmatrix}$$

Rule D2 If an entity can be linked by a relation to another one, if it is optional, i.e., it may not exist at all, and if it is dependent on the other one, i.e., it cannot exist without it, then it is in a direct relation with the relation.

$$\begin{pmatrix} relation_t(R, T_r) \wedge \\ role(T_r, Role_1) \wedge \\ role(T_r, Role_2) \wedge \\ Role_1 \neq Role_2 \wedge \\ rolecardmin(Role_1, 1) \wedge \\ cardmax(T_r, Role_1, 1) \wedge \\ cardmin(T_r, Role_2, 0) \wedge \\ cardmax(T_r, Role_2, 1) \wedge \\ rolefillervalue_t(R, Role_1, E_1) \end{pmatrix} \rightarrow \exists D.dir_t(D, E_1, R)$$

For example, in the specification model of appendix A.1, a home page is an optional entity, dependent on the existence of a site. A site is itself an optional entity, dependent on the existence of a research group, a publication set or a researcher set. Therefore, the relations linking together these elements will be mentioned in the same focus space as the entity they depend on. As a result, research group, site, and home page will be grouped together when discussed in a dialogue (see section 7.4).

Rule D3 Relation role-fillers are in direct relations with their associated relation.

$$rolefillervalue_t(R, Role, E) \rightarrow \exists D.dir_t(D, R, E)$$

There are three rules for creating specialisation relations:

Rule S1 If an entity can be linked to several others playing the same role, then there is a specialisation relation between this entity and the relations linking it to the others. This is because we consider this entity more central than the several other entities that can be related to it.

$$\begin{pmatrix} relation_t(R, T_r) \wedge \\ role(T_r, Role_1) \wedge \\ role(T_r, Role_2) \wedge \\ Role_1 \neq Role_2 \wedge \\ rolecardmax(Role_2, C) \wedge \\ C > 2 \wedge \\ rolefillervalue_t(R, Role_1, E_1) \end{pmatrix} \rightarrow \exists S.spe_t(S, E_1, R)$$

For example, in the specification model of appendix A.1, the research group involves the researcher set and the publication set which fill the same role `involving`. The research group is therefore in a specialisation relation with the relations linking it to the researcher set and the publication set.

Rule S2 If an entity can be linked by a relation to another one, and if it is optional but not dependent on the other one, it is in a specialisation relation with the relation. Since the entity can exist on its own, it is not included in the other entity focus space. However, since it is optional, discussing it in the light of the other entity seems a good way of introducing it.

$$\left(\begin{array}{r} relation_t(R,T_r) \wedge \\ role(T_r, Role_1) \wedge \\ role(T_r, Role_2) \wedge \\ Role_1 \neq Role_2 \wedge \\ cardmin(T_r, Role_1, 0) \wedge \\ cardmax(T_r, Role_1, 1) \wedge \\ rolecardmin(T_r, Role_2, 0) \wedge \\ cardmax(T_r, Role_2, 1) \wedge \\ rolefillervalue_t(R, Role_1, E_1) \end{array}\right) \rightarrow \exists S.spe_t(S, E_1, R)$$

Rule S3 Two entities are in a specialisation relation if a relation (R) linking them is in a specialisation relation with one of them.

$$\left(\begin{array}{r} rolefillervalue_t(R, Role_1, E_1) \wedge \\ rolefillervalue_t(R, Role_2, E_2) \wedge \\ Role_1 \neq Role_2 \wedge \\ spe_t(S, E_1, R) \end{array}\right) \rightarrow \exists S'.spe_t(S', E_1, E_2)$$

There are two rules for creating generalisation relations:

Rule G1 Generalisation relations are the reverse of specialisation relations.

$$spe_t(S, E_1, E_2) \rightarrow \exists G.gen_t(G, E_2, E_1)$$

Rule G2 Rule G1 is modified for the entity-relation case: An entity is in a generalisation relation with a relation if the other role filler is in a specialisation relation with this relation.

$$\left(\begin{array}{r} rolefillervalue_t(R, Role_1, E_1) \wedge \\ rolefillervalue_t(R, Role_2, E_2) \wedge \\ Role_1 \neq Role_2 \wedge \\ spe_t(S, E_2, R) \end{array}\right) \rightarrow \exists G.gen_t(G, E_1, R)$$

Finally, there are two rules for creating simple relations:

Rule R1 An entity and a relation are in simple relation if the entity is a role filler of the relation.

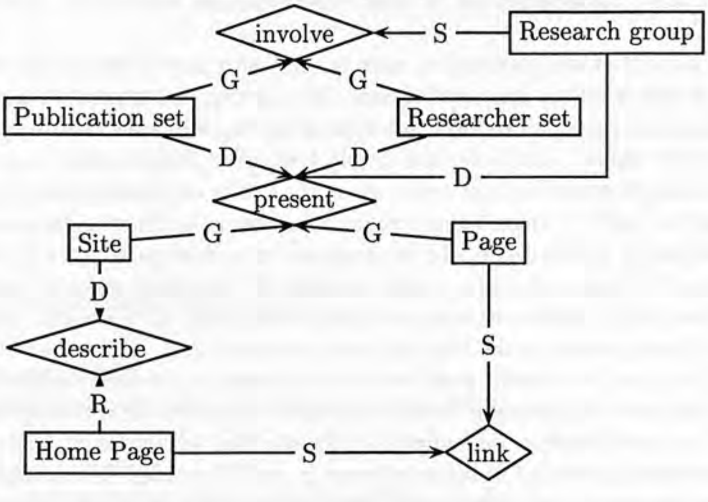
$$rolefillervalue_t(R, Role, E) \rightarrow \exists R.sim_t(R, E, R)$$

Rule R2 Two entities are in simple relation if they are linked by a relation (R).

$$\left(\begin{array}{l} rolefillervalue_t(R, Role_1, E_1) \wedge \\ rolefillervalue_t(R, Role_2, E_2) \wedge \\ Role_1 \neq Role_2 \end{array}\right) \rightarrow \exists R.sim_t(R, E_1, E_2)$$

Rule D1 is somewhat different from the other rules since it depends on the current focus spaces. On the other hand, it is independent from the nature of the relation considered: If two entities are in the same focus space, any relation linking them is in direct relation with them, irrespective of what this relation is. This is useful, for example, in case users mention two entities together which were a priori not expected to be related.

Figure 4.6 Direct, specialisation, generalisation and simple relations



The figure represents the by-default direct (D), specialisation (S), generalisation (G) and simple (R) relations from entities to relations (arrow direction). There is always a direct relation from relations to entities (opposite to arrow direction; translation rule D3). The relations may change as the dialogue evolves, depending on what is in focus (focus spaces general properties and translation rule D1).

Basing the translation rules on the structure of the specification model has the advantage of not being dependent on the semantics of the relations contained in the specification which is a major problem for some focus theories. We are still dependent on the semantics of the specification model. However, this dependency is at a higher level than domain specific ones. It is therefore usually easier to adapt the system to new domains. This advantage can however be a drawback for some domains where the structure of the ER model alone may not be sufficient to guide the dialogue evolution efficiently. In that case, we may need to take into account more domain specific information when carrying out the translation. Reaching a balance between domain dependence and translation power is a difficult task.

Changing the translation rules gives the theory a more local or more global aspect. If more direct relations are generated, entities and relations will be grouped into a few large focus spaces. If more specialisation relations are generated, entities and relations will be grouped in numerous small focus spaces. In our case, the former case is preferred as local focus decisions are the task of a local focus theory (see section 3.3). Choosing the right size for focus spaces should ideally depend on the study of existing dialogues. Although we did not study this issue in detail, the choice we made seems to produce dialogues equivalent in structure to some human-to-human dialogues we recorded in the same domain.

The result of applying the translation rules to the simple specification model presented in figure 4.4 and appendix A.1 is presented in figure 4.6. (In case more than one translation is possible, e.g., an entity is in direct and specialisation relation with another, we only show one translation, preferring translations in the following order: direct relations, specialisation relations, generalisation relations and simple relations.) The rules are mainly based on the cardinality restrictions that exist in the specification model. For example, the research group is in a specialisation relation with the researcher set because the research group involves other sets, e.g., publication set, so that rolecardmax(involve,C), C >= 2 and translation rule S1 applies.

Note that the translation rules are not only dependent on the representation frameworks used but also on the application. For example, because we never mention attributes of entities or relations as main subjects of communications, we do not need to have translation rules for them. Attributes are simply treated as independent from one another in our system and are presented in any order when the entity or relation they depend on is in focus. (Any `ask-value(X | Attribute)` communication is in direct relation with X.) This could be a problem if attributes could be grouped into clusters when discussed in the dialogue. For example, some attributes could deal with colour of objects (background colour, border, colour, etc.) while others could deal with their size (length, height, etc.). Mixing these attributes would make the text less coherent than if they were discussed in two clusters. In this case, we would need to put attributes as the main subjects of communications and to organise the dialogue based on their properties. The translation rules would then have to be more domain dependent to classify the attributes in their clusters since this is not indicated by the information present in an ER model. The translation rules proposed here have however been sufficient to deal with various application domains (see chapter 8).

4.5 Operating overview

Given our system's architecture (see figure 4.2) the process driving the system follows algorithm 4.1. This high-level algorithm is detailed in the next chapters and a full version is presented in section 7.1. The algorithm is composed of three basic parts:

- The production of communications by the elicitation module;
- The interpretation of answers by the elicitation module;
- The selection and generation of communications in natural language by the dialogue manager.

algorithm 4.1 System algorithm (overview)

```
do in any order
  task 1 (Elicitation module)
    Produce communications into the Communication Pool
  end task 1
  task 2 (Elicitation module)
    Interpret answers from the Communication Pool
  end task 2
  task 3 (Dialogue manager)
    Select a communication
    Output this communication
    Put the user's answer in the communication pool
  end task 3
end do in any order
```

Chapter 5

Generating Outputs

Objectives

The objectives of this chapter are to:

- Present how the system outputs communications,
- Describe the use of focus theories in this process,
- Present the role of the elicitation system in producing communications,
- Present the role of the focus theories in selecting a communication to output,
- Present the use of templates and anaphora and cue words generation in the natural language generation process.

5.1 Introduction

The generation of communications is the main function of our system. This is done to ask for requirements to fill in the specification, and to provide guidance in the elicitation process. The generation of communications is a cooperative process between the elicitation module and the dialogue manager. Reasoning and communicating are intimately coupled such as in TRAINS (Traum et al. 1994; Ferguson et al. 1996; Ferguson and Allen 1998) for example. The elicitation module influences the dialogue manager by adding communications to the communication pool. The dialogue manager influences the elicitation module by deciding at which point in the dialogue the communications are output. If the elicitation module follows a strict and narrow way of working, always producing a unique communication, then the dialogue manager becomes irrelevant. On the other hand, putting several communications in the communication pool entails that their treatment can be deferred for an unknown amount of time, until the dialogue manager decides to output them. As a result, the elicitation module should balance the advantages and drawbacks of putting too few or too many communications in the communication pool, and the dialogue manager should balance the advantages and drawbacks of more or less delaying the output of communications. This approach is different from systems where the natural language interface is used as an input or output system but does not directly influence the way of working of the system (see section 2.6). The process generating outputs in our system follows algorithm 5.1. Focus

algorithm 5.1 Generating outputs

```
do in any order
  task 1 (Elicitation module)
    Produce communications into the Communication Pool
  end task 1
  task 2 (Dialogue manager)
    Select a communication from the Communication Pool
    Output this communication
  end task 2
end do in any order
```

theories play a predominant role in this process. They are used to decide which communication should be output next to ensure a coherent dialogue. They are also used to improve the quality of the natural language text produced by the system.

5.2 Producing communications

The elicitation module produces two kinds of communications (see section 4.3):

Presentations Presentation communications are used to make users aware of changes in the specification.

Questions Question communications are used to ask confirmations or refusals from users of changes which could be made in the specification.

We now describe these two kinds of communications in more detail.

5.2.1 Presentation communications

Presentation communications are used to inform users of changes in the specification. Therefore, they are produced:

- whenever the elicitation module automatically asserts new entities or relations in the specification. This happens when the model constraining the specification imposes the existence of some elements. For example, the model presented in appendix A.1 imposes the existence of a research group. Therefore, the elicitation module asserts this entity in the specification and produces a presentation communication to inform users of the change.
- whenever inputs from users lead to the assertion of a new entity or relation in the specification (see chapter 6). In this case, the elicitation system produces presentation communications. These communications are used to confirm the changes to the users and therefore serve as a means of feedback.

If a presentation communication presents something more informative than another one about the same content, e.g., a communication states that an entity is filling a role while another only states that the entity exists, then the less informative communication is not output. The dialogue manager considers it as having already been output. In order to know whether a present communication, C_1, is less informative than another, the dialogue manager looks for another present communication, C_2, with the same main subject and more subjects than the C_1. If such a communication exists, C_1 is considered less informative than C_2. There is no reasoning involved about the meaning of the communication. It would be interesting to cast this ad-hoc procedure in terms of well-known theories like Grice's maxims (Grice 1975). Because C_1 and C_2 share the same main subject, they should be relevant at roughly the same points in the dialogue. However, we prefer to output C_2 rather than C_1 because of the Quantity maxim: "Make your contribution as informative as required (by the current purpose of the exchange)". Just saying that an entity exists may be misleading, if we already know that it is the role filler of some relation. Users may search for an explanation of why we stressed the existence of the entity.

5.2.2 Question communications

Question communications are produced to ask if some changes should be made in the specification. This happens when the changes are not mandatory, otherwise they would be performed automatically (see section 5.2.1), when the changes do not violate the constraints of the model underlying the specification, otherwise they would be blocked, and when the changes have not already been denied by users, otherwise they would again be blocked (see section 6.4.1). There are three kinds of questions which are associated with different transformations of the specification (see section 4.3). They are described below in terms of the elicitation module basic actions (see section 4.2.2).

Ask relation creation (`ask-creation`) These communications are produced whenever a new relation could be asserted. An hypothetical relation is created by basic action 1b, as well as its associated roles by basic action 1c. A question about their validity is added to the communication pool by basic action 2b. In most cases one of the roles of the hypothetical relation is filled by an existing entity while the other is filled by an hypothetical entity created by basic action 1a. This corresponds to the situation where the elicitation module wants to create a new relation from an existing entity to another one, the other being unknown. The identity of this unknown entity will then have to be determined by a `ask-role-filler` question. In some cases, the two entities filling the relation roles are known and the second role-filler is an existing entity. A typical state of the system and its transformation by these actions are presented in table 5.1.

Ask role-filler (`ask-role-filler`) These communications are produced by basic action 2c whenever the role-filler of a role is unknown. This can be the case when a new relation is asserted but one of its roles is not yet filled as explained above. A typical state of the system and its transformation by these actions are presented in table 5.2.

Ask attribute value (`ask-value`) These communications are produced by basic action 2d whenever the value of an attribute is unknown. This happens whenever a new entity or relation is asserted which has attributes to be filled. A typical state of the system and its transformation by these actions are presented in table 5.3.

The creation of new entities is not asked about directly. New entities are created as new role-fillers of relations.

Table 5.1 Ask relation creation specification modifications

<table>
<tr><td>Specification
entity(e1, entity-set1)

Communication pool
</td><td>→</td><td>Specification
entity(e1, entity-set1)
?entity(?e2, entity-set2)
?relation(r1, relation-set1)
?rolefillervalue(r1, role1, e1)
?rolefillervalue(r1, role2, ?e2)

Communication pool
ask-creation(r1 | e1, entity-set2)</td></tr>
</table>

Table 5.2 Ask role-filler specification modifications

<table>
<tr><td>Specification
entity(e1, entity-set1)
?entity(?e2, entity-set2)
relation(r1, relation-set1)
rolefillervalue(r1, role1, e1)
rolefillervalue(r1, role2, ?e2)

Communication pool
</td><td>→</td><td>Specification
entity(e1, entity-set1)
?entity(?e2, entity-set2)
relation(r1, relation-set1)
rolefillervalue(r1, role1, e1)
rolefillervalue(r1, role2, ?e2)

Communication pool
ask-role-filler(r1 | role2, entity-set2)</td></tr>
</table>

5.2.3 Example

Assume that the specification is based on the model described in appendix A.1. Assume also that the system is in the state presented in table 5.4. Because an instance of type `researcher set` must exist, it is created automatically by the system. A presentation communication is added to the communication pool to inform users of this creation. Then, because this new instance must be linked to a research group and there is only one research group allowed by the specification model, an `involve` relation is created and its roles filled by the researcher set and the research group. A presentation is added to inform users of this new relation. Because this last communication is more informative than the previous one, it replaces the previous communication in the communication pool. The state of the system after these modifications is presented in table 5.5. (Focus set and centres have been added in table 5.5. They are managed independently from the production of communications as

Table 5.3 Ask attribute value specification modifications

<table>
<tr><td>Specification
entity(e1, entity-set1)

Communication pool
</td><td>→</td><td>Specification
entity(e1, entity-set1)

Communication pool
ask-value(e1 | att1)</td></tr>
</table>

Table 5.4 Communication generation (system initial state)

Specification	Communication pool
entity(rg1, research group) entity(?s1, site) relation(p1, present) rolefillervalue(p1, presenting, rg1) rolefillervalue(p1, presented by, ?s1)	present(p1 | rg1) ask-role-filler(p1 | presented by, site)

explained in section 5.3.)

5.3 Selecting a communication

The communication selection process is carried out by the dialogue manager. It follows algorithm 5.2. Each step is described below:

algorithm 5.2 Selecting a communication

if the communication pool contains communications to output **then**
  Collect the active communications from communication pool
  while There is no single best ranked communication and there are formal theories to apply **do**
    Rank the up-to-now best ranked communications with the next theory
  end while
  Select best ranked communication
  Update focus state
else
  Stop
end if

Collection This step simply collects the communications in the communication pool which have not already been output. These communications are called "active" communications. If the communication pool does not contain any active communications to output, the system stops.

Ranking The dialogue manager uses the formal dialogue theories (see chapter 3) to select a communication. This is done by using the translation rules to interpret the communication in terms of the notions used by the focus theories, checking which focus rules can be triggered by the communications and then using the ranking on the rules. Because theories ranking the communication earlier in the process define the space in which theories applied later in the process operate, theories should be applied in order of decreasing importance. For example, if we apply them in order global focus theory (see section 3.2), local focus theory (see section 3.3), and present-first theory (see section 3.4), the algorithm will pick the best communication for present-first *among* the best communications for local focus *among* the best communications for global focus. In this example, the global focus theory is therefore more important than the local focus theory and the present-first theory. The former is, in turn, more important than the latter. An example of how the ranking process is carried out in practice is given below (see section 5.3.1).

The order in which the theories are applied can be changed as the dialogue evolves. For example, if the dialogue goes too deeply into a particular direction (as counted, for example, by the number of focus spaces controlling the active one), it is possible to change the ranking policy to correct this. In this example, such a correction could be to revert from the local focus theory we use to the initial centering theory where retaining local focus moves are preferred over smooth-shift local moves (see section 3.3). This would make it harder to shift deeper to a new topic. Adaptative focus theory for generation is left as an open issue for further research (see section 11.2.1).

Selection Once communications have been ranked, the best communication is selected. If several communications are on-par, one of them is chosen at random. The communication selected is marked as having been output as this point. This prevents the dialogue manager from outputting the same communication more than once. (Note that we do not show communications that have been output when presenting the communication pool to avoid cluttering it.)

Focus state update Once the selection has been made, the dialogue manager updates the focus state based on the focus moves associated with the chosen communication. This is done by using the action part of the focus rules that correspond to the moves.

Because the content of the communication pool can be changed at any point by the elicitation module (see section 4.2.2), the dialogue manager cannot plan reliably the dialogue ahead (see section 2.6.2) since some existing communications may not have to be output anymore whilst new ones could appear. If the dialogue manager was planning the output by relying on the presence (or absence) of some communications in the communication pool, there would always be a risk that the assumptions made would be violated before the execution of the plan is finished. However, by keeping track of the focus, the dialogue manager can still ensure a coherent dialogue without planning (Sibun 1992). Depending on the focus theories used and on their ordering, an emphasis is put on ensuring global or local coherence.

5.3.1 Example

Assume that the specification is based on the model described in appendix A.1 and that the system current state is that of table 5.5. Assume also that we are using the present-first, local focus and global focus theories (from less important to more important). The first step for selecting a communication to output is to collect the communications from the communication pool. This gives us:

1. present(i1 | rs1, rg1),
2. present(p1 | rg1),
3. ask-role-filler(p1 | presented by, site).

The second step consists of ranking these communications, by calling the theories in decreasing importance order. The global focus theory is therefore applied first. Communications 2 and 3 are associated with a no change global focus move since relation #p1 is in

Table 5.5 Communication selection (system initial state)

Specification	Communication pool
entity(rg1, research group)	present(i1 | rs1, rg1)
entity(rs1, researcher set)	present(p1 | rg1)
entity(?s1, site)	ask-role-filler(p1 | presented by, site)
relation(i1, involve)	**Focus set**
relation(p1, present)	$\mathcal{F}_1 =_2 \{rg1, p1\}$
rolefillervalue(i1, involving, rs1)	**Centres**
rolefillervalue(i1, involved in, rg1)	$Cf_2(p1, rg1)$
rolefillervalue(p1, presenting, rg1)	$Cp_2(p1)$
rolefillervalue(p1, presented by, ?s1)	$Cb_2(rg1)$

direct relation with itself. On the other hand, communication 1 is associated with an additive global focus move because a specialisation relation exists between research group #rg1 and relation #i1 (see figure 4.6). The first ranking is therefore (2, 3) > 1. The second theory to be applied is the local focus theory. It is applied to the up-to-now best ranked communications. Both communication 2 and communication 3 are associated with a smooth-shift local focus move since their backward-looking centre is different from the current one but is equal to their preferred centre (see table 5.6). The ranking after the application of the local focus theory is therefore unchanged. The present-first theory is applied last. This theory gives us the ranking 2 > 3, since presentation communications are given a better ranking than other communications. In the end, the best ranked communication is communication 2. This is summarised in table 5.6.

Table 5.6 Communication ranks

Communication	Global focus move $\mathcal{F}_1 =_2 \{rg1, p1\}$	Local focus move $Cb_2(rg1), Cp_2(p1), Cf_2(p1, rg1)$
1. present(i1 | rs1, rg1)	additive $spec_2(\_, i1, rg1)$	retaining $Cb_2(1, rg1), Cp_2(1, i1)$
2. **present(p1 | rg1)**	no change $dir_2(\_, p1, p1)$	smooth shift $Cb_2(2, p1), Cp_2(2, p1)$
3. ask-role-filler(p1 | presented by, site)	no change $dir_2(\_, p1, p1)$	smooth shift $Cb_2(3, p1), Cp_2(3, p1)$

Global and focus spaces are reminded at the top of the table. Elements of no interest are replaced by underscores. The selected communication is set in bold face. The reasons for the focus moves of the selected communication are given in each focus move cell. The notation used is explained in sections 3.2 and 3.3.

Communication 2, `present(p1 | rg1)`, is selected in the third step of the algorithm.

The focus state is then updated. Global focus does not change since the move associated with the communication is a no change move and relation #p1 and research group #rg1 are already in the active space. Local focus on the other hand does evolve. The move associated with the communication is a smooth-shift. Therefore, we get $Cp_3(p1)$ and $Cb_3(p1)$. The forward-looking centres are now relation #p1 and research group #rg1. The final state of the system is summarised in table 5.7.

Table 5.7 Communication selection (system final state)

Specification
entity(rg1, research group)
entity(rs1, researcher set)
entity(?s1, site)
relation(i1, involve)
relation(p1, present)
rolefillervalue(i1, involving, rs1)
rolefillervalue(i1, involved in, rg1)
rolefillervalue(p1, presenting, rg1)
rolefillervalue(p1, presented by, ?s1)

Communication pool
present(i1 | rs1, rg1)
ask-role-filler(p1 | presented by, site)

Focus set
$\mathcal{F}_1 =_2 \{rg1, p1\}$

Centres
$Cf_3(p1, rg1)$
$Cp_3(p1)$
$Cb_3(p1)$

5.4 Outputting a communication

The generation of a communication in natural language mainly consists of translating the notation used by the elicitation module into natural language and of ensuring a grammatically correct output corresponding to the communication's intention. We simplify this process by choosing a direct correspondance between the elicitation module's language and natural language and by using templates defining the output sentence's structure. Using templates for generation is a simple but effective technique for unsophisticated outputs (Reiter 1995). For each type of communication, templates define the basic structure of the sentence that will be output based on the specification's current state. A template is chosen based on focus information. Then, the rest of the sentence is filled in by taking into account the information to be output. Although the natural language generation capabilities of our system are limited as there is no point in developing a complex generator for our application, we take into account focus related phenomena and we illustrate how they can influence the generation process. These phenomena would have the same effects in more complex generators. In particular, the current state of the dialogue is considered for template selection (see section 5.4.1) and anaphora and cue word generation (see sections 5.4.2 and 5.4.3). The overall process is described in algorithm 5.3.

algorithm 5.3 Outputting a communication

Select a template based on the communication to be output, the current state of the dialogue and the current specification
Generate anaphora based on the current specification, the current state of the dialogue and the focus moves done to select the communication
Generate cue words to introduce the sentence based on the focus moves done to select the communication
Present the sentence to users

The steps of algorithm 5.3 are now described in more detail.

5.4.1 Template selection

Templates are composed of three parts: a part indicating for what kind of communication they can be used, a part indicating what should be present in the specification for the template to apply and a part giving the structure of the sentence that will be generated. The access to the specification is needed since the dialogue manager does not keep a separate record of what elements in the communication refer to (see section 4.4). The templates used in the system are presented in tables 5.8 and 5.9. The sentence structure part of templates are represented as lists in Prolog style. The variables refer to elements in the specification.

Table 5.8 Presentation output generation templates

Communication	*Specification*
present(Entity1)	entity(Entity1, Entity-set1)
Sentence structure	
[Entity1, is a, Entity-set1]	
Communication	*Specification*
present(Relation1)	entity(Entity1, Entity-set1)
	entity(?Entity2, Entity-set2)
Sentence structure	relation(Relation1, Relation-set1)
[Entity1, Role2, Entity-Set2] (or)	rolefillervalue(Relation1, Role1, Entity1)
[Entity-Set2, Role1, Entity1]	rolefillervalue(Relation1, Role2, ?Entity2)
Communication	*Specification*
present(Relation1)	entity(Entity1, Entity-set1)
	entity(Entity2, Entity-set2)
Sentence structure	relation(Relation1, Relation-set1)
[Entity1, Role2, Entity2] (or)	rolefillervalue(Relation1, Role1, Entity1)
[Entity2, Role1, Entity1]	rolefillervalue(Relation1, Role2, Entity2)

By default, the variables in the sentence structure part of templates are replaced when the sentence is output. Entity-set, relations and attributes are replaced by the name of the element they stand for in the specification. For example, if the specification model contains `entity-set(researcher)` and if `researcher` is assigned to Entity-set1, then Entity-set1 would be replaced by "researcher" in the output. Entities are replaced by the entity-set name and the identifier of the element they stand for. For example, if the specification contains `entity(r1, researcher)` and if r1 is assigned to Entity1, then Entity1 would be replaced by "researcher r1" in the output. The complete template [Entity1, is a, Entity-set1] would be output as "researcher r1 is a researcher". There is one extension to this naming process: if an entity has a `name` or `title` attribute, the value of this attribute is added to the name of the entity. Finally, roles are replaced by conjugated verbs or present or past participles. Conjugated verbs are in the present tense either in active or passive mood. The decision on how to output roles is carried out by an ad-hoc mechanism depending on the name of the roles and the template used.

This very simple generation process has limitations:

- Some n-ary relations ($n \geq 3$) cannot be dealt with properly (see section 8.3). It is sufficient, however, to demonstrate the natural language generation process in most cases we deal with here.

Table 5.9 Question output generation templates

Communication ask-creation(Relation1 | E1, Entity-set2) *Sentence structure* [Do you want, E1, Role2, Entity-set2, ?] (or) [Do you want, Entity-Set2, Role1, E1, ?]	*Specification* entity(E1, Entity-set1) ?entity(?E2, Entity-set2) ?relation(Relation1, Relation-set1) ?rolefillervalue(Relation1, Role1, E1) ?rolefillervalue(Relation1, Role2, ?E2)
Communication ask-role-filler(Relation1 | Role2) *Sentence structure* [Which, Entity-set2, Role1, Entity1, ?]	*Specification* entity(Entity1, Entity-set1) ?entity(?Entity2, Entity-set2) relation(Relation1, Relation-set1) rolefillervalue(Relation1, Role1, Entity1) rolefillervalue(Relation1, Role2, ?Entity2)
Communication ask-value(Entity1 | Attribute1) *Sentence structure* [Who/What is, Attribute1, of, Entity1, ?]	*Specification* entity(Entity1, Entity-set1)
Communication ask-value(Relation1 | Attribute1) *Sentence structure* [Who/What is, Attribute1, of, Relation1, ?]	*Specification* relation(Relation1, Relation-set1)

- Relations are always output as verbs. This is a problem when a relation is the backward-looking centre or a forward-looking centre of a sentence. Centres in the centering theory should indeed be realised as nouns (see section 2.4.1). For example, rather than generating "The home page is linked to a page", we should generate "A link joins the home page to a page", or something equivalent, when relation `link` is a centre. We could deal with this problem in several ways:
  - Change the generator so that relations can be output as nouns. This may lead however to complex sentences and may not always be possible.
  - Use another focus theory than centering, which allows verbs to be in focus (e.g., Sidner's focus theory (Sidner 1979)). While this solution would solve the particular problem discussed here, it is likely that other problems will appear.
  - Ensure that relations are never centres of sentences. This could be achieved by modifying the translation rules used for the centering theory. However, this approach would be ad-hoc for the generator we use and quite restrictive.

  This problem is in fact a manifestation of a more important one. Our system implements a pipelined architecture: the dialogue manager makes a decision on which communication to output and how it should be output; it then instructs the generator to realise the communication as a sentence. Problems arise when the generator is not able to generate the sentences requested by the dialogue manager. A feedback loop would be necessary for the generator to warn the dialogue manager of potential problems. The two programs would then have to agree on a way of solving them: output the same communication but in a different form or output a different communication. In our system, we did not implement such a loop. The generator tries to realise the sentence as well as it can, respecting the dialogue manager's instructions. If it cannot respect some instructions, they are simply ignored. This way of working does not seem to affect the text perversly. The system's architecture also remains simple. Balancing generation power and system complexity is a difficult issue (Reiter 1994).

The naming process could be easily improved. For example, each communication could be associated with an ontology, as KQML messages are (Labrou 1996). We could then translate from this ontology to natural language. This would avoid the direct mapping between the internal representation and natural language, which makes the current naming process brittle. However, we did not consider that the possible improvements in natural language were sufficient, compared to the added complexity of using an ontology, for our application.

When several templates can be used, we select the one that will make it easier to recognise the local focus move made to select the communication (see section 3.3.1). In particular, we prefer templates that put the preferred centre of the communication in subject position as suggested by the ordering on forward-looking centres in the centering theory (see sections 2.4.1 and 3.3.1). This may for example involve the use of a passive sentence rather than an active one when presenting relations. Since we deal with binary relations, this corresponds to describing a relation from the point of view of one entity rather then from the point of view of the other entity the relation links. (This is equivalent to what is done in McKeown (1985a, p. 78).) This is shown in the example below.

Example

Given the state of the system presented in table 5.5, communication `presentation(p1 | rg1)` triggers templates [rg1, presented by, site] and [site, presenting, rg1]. The first template would be output as "Research group #rg1 is presented by a site". However, since the preferred centre is relation #p1 and not research group #rg1, the second template is selected. This template would then be output by default as "A site presents research group #rg1". In this case we prefer the active sentence to its passive counterpart.

5.4.2 Anaphora generation

The system uses the current state of the dialogue to improve the output of the system by replacing the by-default naming process. Two rules are used for this purpose:

Rule 5.1 (Global definite noun phrase generation) *If a variable refers to an entity set, or an instance of an entity set, which can only have one instance, a definite phrase can be used.*

For example, assume that the specification is based on the model described in appendix A.1 and that it contains `entity(rg1, research group)`. Assume further that rg1 is assigned to Entity1. Then Entity1 will be output as "*the* research group" since there can only be one research group in the dialogue.

This rule is very simple. Better definite noun phrases could be produced by taking into account focus information (see section 6.1 for an example of such definite noun phrases resolution). However, the production of definite noun phrases is a complex problem (Dale 1992) and the solution adopted here, although simple, performs reasonably well.

Definite phrases are not generated when presenting entities. Therefore, when outputing `present(rg1)`, we still generate "research group #rg1 is the research group" rather than "the research group is the research group", which would not make much sense.

Rule 5.2 (Focused pronoun generation) *A pronoun can be used to name an entity if it is the new backward-looking centre of the dialogue and the local focus move done to select the communication is a continuation or retaining move.*

This rule enables the system to generate pronouns for entities which are the most important ones in the current sentence if they were already in focus in the last utterance. This rule may trigger further transformations to ensure the readability of a sentence. In particular, the output "...of it" is transformed into "its ...".

This rule is more stringent than the rule proposed in the centering theory (see section 2.4.1) since we impose that the element to be realised as a pronoun be the backward-looking centre of the current and previous sentence. It corresponds to the rule given in Grosz et al. (1983, p. 48):

> If the Cb of the current utterance is the same as the Cb of the previous utterance,
> a pronoun should be used.

In centering theory, the element realised as a pronoun must also be the backward-looking centre of the current sentence, but just needs to be a member of the previous sentence forward-looking centres. Our more stringent rule enables us to make a clearer distinction between continuation and retaining local focus moves, where the same element remains in focus and can be realised by a pronoun, and smooth-shift and rough-shift local focus moves, where the backward-looking centre changes. Because the rule is more stringent than the centering theory rule, the pronouns it produces are a subset of possible pronouns and should therefore be easily interpreted by users. This rule is not used if local focus information is not available.

When both anaphora generation rules can apply, a pronoun is produced rather than a definite noun.

Example

A definite noun phrase can be generated for the template presented in the example above since research group #rg1 is the only research group that can exist. On the other hand, a pronoun cannot be generated because research group #rg1 is not the backward-looking centre of the sentence (which is #p1, $Cb_3(p1)$). The output is therefore "A site presents the research group".

5.4.3 Cue word generation

Cue words have been recognised as playing a major role in dialogue. In particular, they help the dialogue interpretation process by pointing out focus shifts. However, little work has been done in determining when cue words should be used to mark focus shifts and how to select them. Cohen (1984, 1987) proposes that cue words are used:

- to specifically re-direct the attention of the hearer to an earlier part of the dialogue. An example of such a cue word is *returning to*.
- to indicate how the following utterances are related to the current focus. Examples include *first* or *second* to introduce a parallel focus and *in particular* to introduce a more specific subject.

Cohen then states that these two types of cue words can be used either to ease the dialogue interpretation, in which case they are optional, or to allow an interpretation that would be otherwise impossible, in which case they are mandatory. An interpretation is considered to be impossible without cue words if another one exists which requires less effort to understand. (The calculation of the amount of effort needed to interpret a dialogue depends on the dialogue theory used.) The further the intended interpretation is from the least costly one, the stronger the use of cue words needs to be.

This issue was also alluded to by Linde and Goguen (Linde and Goguen 1978). In particular, they propose to classify the strength of cue words on the following criteria:

- The more a cue word looks like a sentence, i.e., the more independent it is, the stronger it is.
- Cue words in front of sentences are stronger than those placed within sentences.
- The longer and more explicit a cue word is, the stronger it is.

For example, the cue in the first sentence of table 5.10 is stronger than the cue in the second. It can then be used to mark a bigger deviation from the expected flow of discourse.

Table 5.10 Cue word strength (from Linde and Goguen (1978, p. 238))

Strong cue word: On the other hand, we have to...
Weak cue word: We also have to...

In spite of these pieces of work, the use of cue words, especially in text generation, is still not fully understood. Systematic work like Moser and Moore (1996) and Knott and Dale (1996) may shed light on this issue.

In our system cue words are generated for:

- global focus pops (see section 3.2). More specifically, a communication is introduced by "Let's come back to our previous topic" if the global focus pop rule has been triggered at least twice to select it, i.e., when we go up in the controlling space hierarchy above the parent level.
- IRU moves (see section 3.4). More specifically, the presentation communication associated with the move is output as "Remember that..." as an introduction to the main communication. For example, if the system has been speaking about the researcher set and then discusses the research group for some time, coming back to the researcher set would trigger cue words such as "Remember that researcher set #rs1 is involved in research group #rg1". IRU moves are often done at the same time as pop moves. It is therefore usual to combine both move cue words into "Let's come back to our previous topic. Remember that ..."

Focus movement is also used in Moeller (1996) to produce sentences making the text flow explicit. In this piece of work, several sentences can be produced in order to make explicit how the focus evolves.

5.4.4 Sentence presentation

The final step in the process of outputting a communication is to present it to users. Then, if a presentation communication has been output, the dialogue manager immediately outputs another communication. If a question communication has been output, a prompt for answer is displayed to users and the control is given to the interpretation process (see chapter 6).

5.5 Examples

We present now some examples of natural language generation for different communication types under different conditions.

5.5.1 Present communication

Table 5.11 Generating natural language for `present(p1 | rg1, site)`

Specification	
entity(rg1, research group)	rolefillervalue(p1, presenting, rg1)
entity(?s1, site)	rolefillervalue(p1, presented by, ?s1)
relation(p1, present)	

Focus set	Centres
$\mathcal{F}_1 =_2 \{rg1, p1\}$	$Cf_2(p1, rg1)$
	$Cp_2(p1)$
	$Cb_2(rg1)$

communication: `present(p1 | rg1, site)`

Given the state of the dialogue presented in table 5.11, communication `present(p1 | rg1, site)` corresponds to a no change global focus move since #p1 is in the current active focus space and is in direct relation with itself, and a smooth-shift local focus move since the backward-looking centre of the communication is #p1 which is its preferred centre as well.

The communication can be output by either template [Entity1, Role2, Entity-Set2] or template [Entity-Set2, Role1, Entity1]. Because relation #p1 is the preferred centre and not research group #rg1, the second template is selected since the first one would put research group #rg1 in a subject position which we want to avoid. Entity-Set2 then contains `site`, Role1 contains `presenting` and Entity1 contains #rg1.

We can now replace the variables in the template by their natural language equivalent. `site` is simply replaced by "A site", `presenting` is replaced by the present form "presents". Because research group #rg1 is the only research group that can exist in the specification, rule 5.1 applies and it is replaced by "the research group".

The sentence is then output as "A site presents the research group".

5.5.2 Ask-creation communication

Given the state of the dialogue presented in table 5.12, communication `ask-creation(p1 | rg1, site)` corresponds to a no change global focus move and a retaining local focus move. The no change global focus move is due to the direct relation between research groups and

Table 5.12 Generating natural language for `ask-creation(p1 | rg1, site)`

Specification	
entity(rg1, research group)	rolefillervalue(p1, presenting, rg1)
entity(?s1, site)	rolefillervalue(p1, presented by, ?s1)
relation(p1, present)	

Focus set	Centres
$\mathcal{F}_1 =_1 \{rg1\}$	$Cf_1(rg1)$
	$Cp_1(rg1)$
	$Cb_1(rg1)$

communication: `ask-creation(p1 | rg1, site)`

presentation relations (see figure 4.6). The retaining local focus move is due to the fact that the communication backward-looking centre is research group #rg1 which is already the backward-looking and its preferred centre is relation #p1.

Two templates can be used to output ask-creation communications: [Do you want, Entity1, Role2, Entity-set2, ?] and [Do you want, Entity-Set2, Role1, Entity1, ?]. Because relation #p1 is the preferred centre and not research group #rg1, the second template is selected since the first one would put research group #rg1 in a subject position. Entity1 is then played by #rg1, Role1 by `presenting` and Entity-set2 by `site`.

We can now replace the variables in the template by their natural language equivalent. `site` is simply replaced by "A site", `presenting` is left as the present participle form "presenting". Because research group #rg1 is the backward-looking centre of the communication and the local focus move associated with the communication is a retaining move, rule 5.2 applies and a pronoun is generated.

The sentence is then output as "Do you want a site presenting it?".

5.5.3 Ask-role-filler communication

Table 5.13 Generating natural language for `ask-role-filler(p1 | presented by, site)`

Specification	
entity(rg1, research group)	rolefillervalue(p1, presenting, rg1)
entity(?s1, site)	rolefillervalue(p1, presented by, s1)
relation(p1, present)	

Focus set	Centres
$\mathcal{F}_1 =_3 \{rg1, p1\}$	$Cf_3(p1, rg1)$
	$Cp_3(p1)$
	$Cb_3(p1)$

communication: `ask-role-filler(p1 | presented by, site)`

Given the state of the dialogue presented in table 5.13, communication `ask-role-filler(p1 | presented by, site)` corresponds to a no change global focus move and a continuation local focus move. The no change global focus move is due to the presence of #p1 in the focus space and to the direct relation of #p1 with itself. The continuation local focus move comes from the fact that the communication preferred centre and its backward-looking centre are both #p1 which is already the backward-looking centre.

Only the template [Which, Entity-set2, Role1, Entity1, ?] can be used to express the communication. Entity-set2 is played by `site`, Role1 is played by `presenting` and Entity1 is played by #rg1.

As already explained in the previous examples, `site` is replaced by "site", `presenting` is replaced by "presents" and #rg1 is replaced by "the research group". No pronoun generation is possible since research group #rg1 is not the backward-looking centre of the communication.

The sentence is then output as "Which site presents the research group?".

5.5.4 Ask-value communication

Table 5.14 Generating natural language for `ask-value(hp1 | title)`

Specification
entity(hp1, home page)

Focus set
$\mathcal{F}_1 =_7 \{rg1, p1, s1, d1, hp1\}$

Centres
$Cf_7(d1, s1, hp1)$
$Cp_7(d1)$
$Cb_7(d1)$

communication: `ask-value(hp1 | title)`

Given the state of the dialogue presented in table 5.14, communication `ask-value(hp1 | title)` corresponds to a no change global focus move and a smooth-shift local focus move. The no change global focus move is due to the presence of #hp1 in the focus space and to the direct relation of #hp1 with itself. The smooth-shift local focus move comes from the fact that the communication preferred centre and its backward-looking centre are both `p1` which is not the current backward-looking centre.

Only template [What is, Attribute1, of, Entity1, ?] can be used to express the communication. Attribute1 is played by `title` and Entity1 by #hp1. `Title` is realised by "title" and #hp1 is realised by "home page #hp1" since neither a definite phrase nor a pronoun can be used.

The sentence is then output as: "What is the title of home page #hp1?".

5.6 Influencing the output generation

Users can influence whether communications are output or not. They have two ways of doing this: (1) asking the dialogue manager to answer some questions for them or (2) taking the initiative.

5.6.1 Automatic answering

It is quite usual in a specification that all instances of an entity-set share a property. For example, in our WWW site design example, all the pages of a site may be linked to the home page of the site. There is no way of representing this information directly in an entity-relationship model. It is necessary to assert each individual relation. Doing this can be tedious in an elicitation dialogue as soon as the number of entities becomes important. To alleviate this burden, we provide a mechanism to answer some questions automatically. Once users have stated which questions should be answered directly (see section 6.4.4), the dialogue manager answers them without outputting them. The algorithm driving the generation process is therefore slightly modified to take this possibility into account. The new algorithm is algorithm 5.4. Feedback informing users that modifications of the specification

---

algorithm 5.4 Generating outputs with automatic answering mechanism

---

repeat
  Select a communication
  while an automatic answer can be provided **do**
    Answer the question directly
    Select a new communication
  end while
  Output the communication
until the communication is not a presentation communication

---

have been made are put in the communication pool. The dialogue manager will present them when allowed by the state of the dialogue.

5.6.2 Taking the initiative

The second way users can modify how output are generated is by taking the initiative. By default, the system asks questions and users answer them. This way of working may be problematic if users want to input a lot of new information. This may be the case if, for example, they do not need the help of the system to input a large part of the specification. In that case, users should take the initiative. Our system enables users to do this. In its user-initiative mode, the system does not output communications anymore (except for feedback). It just generates new communications and waits for users to provide the requirements they want to enter (see section 6.4.5. An example of such a dialogue is presented in table 6.18 in the same section.). The algorithm driving the generation process is therefore slightly modified to take this possibility into account. It is identical to algorithm 5.4 except for a check on the mixed-initiative parameter. If the parameter is set, the generation process is carried out. Otherwise the process is ignored. This parameter is switched on or off by a special input (see section 6.4.6).

These two ways of modifying the system inputs have been limited during users' experiments with the system. However, users were given the possibility of redirecting the dialogue (see section 6.4.3).

be tedious in an elicitation dialogue, especially as the number of entities becomes important. To alleviate this burden, we provide a mechanism to answer some questions automatically. Once users have stated which questions should be answered directly (see section 6.4.4), the dialogue manager answers them without outputting them. The algorithm driving the generation process is therefore slightly modified to take this possibility into account. The new algorithm is algorithm 5.4. For the following user that, modifications of the specification

---

algorithm 5.4 Generating outputs with automatic answering mechanism

---

repeat
  Select a communication
  while an automatic answer can be provided **do**
    Answer the question directly
    Select a new communication
  end while
  Output the communication
until the communication is not a presentation communication

---

have been made are put in the communication pool. The dialogue manager will present them when allowed by the state of the dialogue.

5.6.2 Taking the initiative

The second way users can modify how output are generated is by taking the initiative. By default, the system asks questions and users answer them. This way of working may be problematic if users want to input a lot of new information. This may be the case if, for example, they do not need the help of the system to input a large part of the specification. In that case, users should take the initiative. Our system enables users to do that: in its user-initiative mode, the system does not output communications anymore (except for feedback). It just generates new communications and waits for users to provide the requirements they want to enter (see section 6.4.5). An example of such a dialogue is presented in table 6.18 in the same section.). The algorithm driving the generation process is therefore slightly modified to take this possibility into account. It is identical to algorithm 5.4 except for a check on the mixed-initiative parameter. If the parameter is set, the generation process is carried out. Otherwise the process is ignored. The parameter is switched on or off by a special input (see section 6.4.6).

These two ways of modifying the system inputs have been limited during users' experiments with the system. However, users were given the possibility of influencing the dialogue (see section 6.6.3).

Chapter 6

Interpreting Inputs

Objectives

The objectives of this chapter are to:

- Describe how users' inputs to the system are interpreted,
- Describe the use of focus theories in this process,
- Present the use of templates with anaphora resolution in the natural language understanding process,
- Present the role of the elicitation system in interpreting simple answers,
- Present the role of the dialogue manager in interpreting more complex answers such as new information and dialogue redirection.

6.1 Introduction

Since our system is involved in an interactive elicitation process, it is necessary to provide a mechanism enabling users to input data. This mechanism should at least allow users to answer questions asked by the system. However, allowing users to provide new information that has not been asked for is also important. In particular, it enables users with some experience to by-pass some questions and be more efficient (Smith 1993, 1996). Allowing free input is a difficult task. Interpreting full natural language is indeed very complex. In order to allow users a certain freedom in their dialogue with the system while restricting the problems due to interpreting full natural language, we only interpret a subset of natural language which seems sufficient for inputing requirements in our domain. The system also keeps track of the evolution of focus during input interpretation. This enables the system to provide appropriate guidance related to the concern of its users. This is different from most approaches where focus rules are used to improve the quality of the system's outputs but not to interpret user's answers (see section 2.6).

The interpretation of users' inputs is a cooperative process between the elicitation module and the dialogue manager. It is based on templates. Using templates is a well-know

algorithm 6.1 Interpreting inputs

Preprocess input
Select template
Carry out triggered actions

technique for interpreting simple natural language (Covington 1994). Templates are composed of an input pattern and some associated actions. If an input matches a template input pattern, the actions associated with the template are triggered. Because inputs do not always directly match a template's input pattern, the inputs may need some preprocessing before being usable. The resulting process follows algorithm 6.1.

We will show templates as lists in Prolog style. Variables are therefore represented by names beginning with an upper-case letter. The constraints on the variables are made obvious by their name. For example, the variable Entity1 must contain an entity and the variable Relation1 a relation.

6.2 Preprocessing

The preprocessing step consists of resolving pronouns and definite noun phrases, i.e., replacing these elements by what they stand for. Three rules are used for resolution:

Rule 6.1 (Global definite noun phrase resolution) *If a definite phrase refers to an entity set and the entity set has only one instance, the definite phrase is replaced by this instance.*

For example, assuming that the specification is based on the model described in appendix A.1, the definite noun phrase `the research group` would be resolved by this rule into #rg if #rg is the identifier of the research group since only one instance of research group can exist in the specification.

Rule 6.2 (Focused definite noun phrase resolution) *If a definite phrase refers to an entity set and instances of this entity set are in the current active global focus, the definite phrase is replaced by one of these instances (until one satisfies all the constraints of the interpreting process including those imposed by the elicitation module).*

This rule enables the system to resolve definite noun phrases like `the page`. In this case, rule 6.1 cannot apply because there may be more than one page. However, it is likely that we are speaking about the one in focus (otherwise we would say `page #n`) which is the one proposed by the rule.

Although this rule may not produce a unique referent for a definite noun, the risk of choosing a wrong interpretation is very small because the number of entities of the same type in the active focus space is limited. Moreover additional constraints are imposed on the referent when carrying out the actions triggered by the template.

This rule does not apply if global focus information is not available.

Rule 6.3 (Focused pronoun resolution) *A pronoun can be replaced by one of the forward-looking centres of the preceding sentence (if it satisfies all the constraints of the interpreting process including those imposed by the elicitation module).*

This rule enables the system to interpret sentences like `I want it to be presented by a site` in response to the question "Do you want the research group to be presented by a page?".

This rule is less stringent than the rule for interpreting pronouns proposed in centering theory (see section 2.4.1) because we do not impose the pronoun to be the backward-looking centre of the sentence. This is because we can find out the backward-looking centre during the interpretation process without knowing which elements have been realised by a pronoun (see section 6.4.2). The fact that this rule is less stringent than the one proposed in the centering theory should not create any problems for users, since they have then more freedom in their use of pronouns.

Although this rule may not produce a unique referent for a pronoun, the risk of choosing a wrong interpretation is very small because the number of forward-looking centres in our system is limited (usually two). Moreover additional constraints are imposed by the template in the selection step and while carrying out the actions triggered by the template. In the example given above, the forward-looking centres are the `research group` and the `presented by` relation. However, only the `research group` verifies the additional constraints imposed during the interpretation process (see section 6.4.2). It is therefore selected as the referent of the pronoun. This rule does not apply if local focus information is not available.

6.2.1 Examples

In table 6.1, we present a dialogue and show the rules used during the preprocessing phase to resolve the noun phrases and pronoun references. (These elements are set in bold in the example text). The interpretation of the exchanges is explained in the rest of this chapter and is examined in detail on an example in chapter 7.

6.3 Selection

The selection step consists of matching the input pattern of a template with the user's input. Templates can contain variables. In that case, the variables are assigned a part of the users' input if this part verifies some constraints. Examples of constraints are that the variable should contain the identifier of an entity, or that it should contain a possible value for an attribute. If no pattern directly matches the input, we try again with a part of the input.

Assume that the specification is still based on the model described in appendix A.1 and that there is a template of the form [speak, about, Entity1]. Now let's suppose the user inputs `I want to speak about the research group`. Since this sentence contains a definite noun phrase, the resolution step is carried out and the system tries to replace the definite noun phrase with what it refers to. Since there can only be one research group in our system, the noun phrase refers to it. It is therefore replaced by the identifier corresponding to the research group. The input thus becomes `I want to speak about #rg`. This input does not directly match a template. Therefore, the system searches for a part of the input matching the template. This part is `speak about #rg`. The template can then be applied and its associated action triggered with Entity1 = #rg which satisfies the constraint.

Table 6.1 Preprocessing

S:	Research group #rg1 is the research group. Do you want a site presenting it?	
U:	Yes.	
S:	A site presents the research group. Which site presents the research group?	
U:	A new one.	
S:	site #s1 presents the research group. Do you want a home page describing site #s1?	
U:	Yes.	
S:	A home page describes site #s1. Which home page describes site #s1?	
U:	A new one.	*Create a home page*
S:	Home page #hp1 describes site #s1.	
S:	Do you want a page linked to it?	
U:	I want to speak about **the researcher set**.	*Create a new focus space.*
S:	OK. Researcher set #rs1 is the researcher set. It is involved in the research group. Do you want a page presenting it?	*Rule 6.1 applies since there is only one researcher set*
U:	Yes.	
S:	A page presents the researcher set. Which page presents the researcher set?	
U:	A new one.	*Create a page*
S:	Page #pa1 presents the researcher set. What is the title of page #pa1?	
U:	I want to speak about **the publication set**.	*Create a new focus space.*
S:	OK. Publication set #ps1 is the publication set. It is involved in the research group.	*Rule 6.1 applies since there is only one publication set*
S:	Do you want a page presenting it?	
U:	Yes.	
S:	A page presents the publication set. Which page presents the publication set?	
U:	A new one.	*Create a page*
S:	Page #pa2 presents the publication set. What is the title of page #pa2?	
U:	**The page** is linked to **the homepage**.	*Rule 6.2 applies since there*
S:	OK. Group homepage #hp1 is linked from page #pa2. What is the title of page #pa2?	*exists two pages (#pa1 and #pa2) but only one (#pa2) is in the current global focus*
U:	Publication.	*and Rule 6.1 applies since*
S:	Do you want it linked from another page?	*there is only one home page*
U:	Yes.	
U:	**It** is linked to page #pa1.	*Rule 6.3 applies*
S:	OK.	

Table 6.2 Simple answer templates

Template	[yes]
Description	Accept the creation of a relation
Action	Assert relation
Template	[no]
Description	Refuse the creation of a relation
Action	Remove relation
Template	[Entity]
Description	Propose Entity as the role-filler of a relation
Action	Set role-filler to Entity
Template	[new]
Description	Propose to create a new entity as the role-filler of a relation
Action	Assert new entity as role-filler
Template	[Value]
Description	Propose Value as the value of an entity or relation attribute
Action	Set entity or relation attribute to Value

6.4 Action

Once a template has been selected, the actions associated with it are carried out. Templates are divided into three kinds, depending on the sort of actions they trigger:

1. Templates dealing with simple answers to questions. These templates trigger actions from the elicitation module by modifying the communication pool.

2. Templates dealing with new information. These templates trigger actions from the dialogue manager which in turn result in actions from the elicitation module as above.

3. Templates dealing with dialogue redirection. These templates trigger actions from the dialogue manager.

6.4.1 Interpreting simple answers

Simple answers are answers to questions that can be directly interpreted by the elicitation module. The dialogue manager just finds the template that should be activated. Then the answer is passed to the elicitation module by adding it to the communication pool (see section 4.3). (In the examples we give, we usually do not present answers in the communication pool to avoid cluttering it.) The elicitation module then checks that the answer is acceptable by verifying that the constraints set in the model underlying the specification are respected. If the constraints are not respected, the answer is rejected and the question is put back in the communication pool. If the constraints are respected, the specification is modified which may then lead to changes in the communication pool. The templates interpreting simple answers are presented in table 6.2 with the actions they trigger. Descriptions of the actions carried out by the elicitation module in response to a user's input, in terms of the elicitation module basic actions (see section 4.2) are given below:

Assert relation The hypothetical relation whose existence has been confirmed by the user is asserted by basic action 3b. The hypothetical roles associated with the relation (see section 5.2.2) are also asserted by basic action 3c. This may then lead to the removal from the specification of hypothetical elements which would now violate the specification model constraints if they were asserted. This is done by basic actions 5a, 5b and 5c. The possible communications related to these elements are removed from the communication pool by basic action 6.

A typical state of the system and its transformation by these actions are presented in table 6.3. In this case, the hypothetical status of relation #r1 is removed. As a result the hypothetical status of its role-fillers is also removed since we are know sure they exist. The only point that remains to be elicited is the exact identity of entity #e2 which may be a new entity or an existing one (depending on the users' wish and the specification model constraints).

Table 6.3 Assert relation specification modifications

Specification		Specification
entity(e1, entity-set1)		entity(e1, entity-set1)
?entity(?e2, entity-set2)	→	entity(?e2, entity-set2)
?relation(r1, relation-set1)		relation(r1, relation-set1)
?rolefillervalue(r1, role1, e1)		rolefillervalue(r1, role1, e1)
?rolefillervalue(r1, role2, ?e2)		rolefillervalue(r1, role2, ?e2)

Remove relation The hypothetical relation whose existence has been denied is removed from the specification by basic action 5b. The hypothetical roles and the potential hypothetical role-filler associated with the relation (see section 5.2.2) are also removed by basic actions 5c and 5a respectively.

A typical state of the system and its transformation by these actions are presented in table 6.4. In this case, the existence of relation #r1 has been denied. It is therefore removed from the specification. Its associated role-fillers are also removed.

This answer has also an important influence on the system future behaviour. Because, refusing to create a relation often means that no more entities of a certain type should be created, e.g., saying that no more researchers belong to the researcher set means that all researchers have already been elicited, the system will not output again questions which are answered negatively.

Table 6.4 Remove relation specification modifications

Specification		Specification
entity(e1, entity-set1)		
?entity(?e2, entity-set2)	→	
?relation(r1, relation-set1)		entity(e1, entity-set1)
?rolefillervalue(r1, role1, e1)		
?rolefillervalue(r1, role2, ?e2)		

Set role-filler to entity The system checks that the entity given can effectively fill the role mentioned. If it can, it is asserted as the role-filler by basic action 4a. The hypothetical role-filler created by default to fill the role (see section 5.2.2) is removed from the specification by basic action 5a. This may then lead to the removals of some elements from the specification and the removal of some communications from the communication pool as when asserting a relation.

A typical state of the system and its transformation by these actions are presented in table 6.5. In this case, users decided that role-filler #e3 was the same as entity #e2. Therefore, #?e3 is replaced by #e2 in the specification and the hypothetical entity #?e3 is removed.

Table 6.5 Assert role-filler specification modifications

Specification		Specification
entity(e1, entity-set1) entity(e2, entity-set2) ?entity(?e3, entity-set2) relation(r1, relation-set1) rolefillervalue(r1, role1, e1) rolefillervalue(r1, role2, ?e3)	→	entity(e1, entity-set1) entity(e2, entity-set2) relation(r1, relation-set1) rolefillervalue(r1, role1, e1) rolefillervalue(r1, role2, e2)

Assert new entity as role-filler If allowed by the constraints on the specification, the hypothetical entity filling the role by default (see section 5.2.2) is asserted by basic action 3a. This may then lead to the removals of some elements from the specification and the removal of some communications from the communication pool as when asserting a relation.

A typical state of the system and its transformation by these actions during are presented in table 6.6. In this case, users decided that role-filler #e2 was a new entity. Its hypothetical status is therefore removed.

Table 6.6 Assert new role-filler specification modifications

Specification		Specification
entity(e1, entity-set1) ?entity(?e2, entity-set2) relation(r1, relation-set1) rolefillervalue(r1, role1, e1) rolefillervalue(r1, role2, ?e2)	→	entity(e1, entity-set1) entity(e2, entity-set2) relation(r1, relation-set1) rolefillervalue(r1, role1, e1) rolefillervalue(r1, role2, e2)

Set attribute Basic action 4b assigns the value provided to the attribute.

A typical state of the system and its transformation by these actions during new role-filler setting are presented in table 6.7. In this case, the value of attribute "attribute" is "value".

Table 6.7 Assert attribute value specification modifications

Specification		Specification
entity(e1, entity-set1)	→	entity(e1, entity-set1) attributevalue(e1, attribute, value)

Table 6.8 Interpreting simple answers (system initial state)

Specification
entity(rg1, research group)
entity(rs1, researcher set)
? entity(?s1, site)
? entity(?pa1, page)
relation(i1, involve)
? relation(p1, present)
? relation(p2,present)
rolefillervalue(i1, involving, rs1)
rolefillervalue(i1, involved in, rg1)
? rolefillervalue(p1, presenting, rg1)
? rolefillervalue(p1, presented by, ?s1)
? rolefillervalue(p2, presenting, rg1)
? rolefillervalue(p2, presented by, ?pa1)

Communication pool
present(i1 | rs1, rg1)
ask-creation(p1 | rg1, site)
ask-creation(p2 | rg1, page)

Focus set
$\mathcal{F}_1 =_2 \{rg1, p1\}$

Centres
$Cf_2(rg1, p1)$
$Cp_2(p1)$
$Cb_2(rg1)$

Table 6.9 Interpreting simple answers (system final state)

Specification
entity(rg1, research group)
entity(rs1, researcher set)
? entity(?s1, site)
? entity(?pa1, page)
relation(i1, involve)
relation(p1, present)
rolefillervalue(i1, involving, rs1)
rolefillervalue(i1, involved in, rg1)
? rolefillervalue(p1, presenting, rg1)
? rolefillervalue(p1, presented by, ?s1)

Communication pool
present(i1 | rs1, rg1)
answer(ask-creation(p1 | rg1, site), yes)

Focus set
$\mathcal{F}_1 =_2 \{rg1, p1\}$

Centres
$Cf_2(rg1, p1)$
$Cp_2(p1)$
$Cb_2(rg1)$

Example

Assume that the model underlying the elicitation process is that of appendix A.1. Assume also that the system is in the state shown in table 6.8, and that the last communication output was `ask-creation(p1 | rg1, site)`, i.e., "Do you want the research group to be presented by a site?". If the user answers `yes`, the dialogue manager will add this answer to the communication pool. The elicitation module will then assert the new relation and its associated roles in the specification. This is shown by the removal of the question marks in front of these elements in the specification. The module will also remove the relation about presenting the research group with a page since a research group can be presented by only one entity, site or page, in the model. Finally, the question communication related to this relation is removed from the communication pool. This gives the new state shown in table 6.9.

6.4.2 Interpreting new information

The interpretation of new information is more complex than the processing of simple

algorithm 6.2 Interpreting new information inputs

Preprocess input
Select template
Divide input in simple answers based on the template
repeat
  Find question communication related to a simple answer
  Update the focus state for this communication (see section 5.3)
  Process the simple answer (see in section 6.4.1)
until all simple answers have been processed
Select and output any feedback (see sections 5.3 and 5.4)

answers. Because the elicitation module has a very limited interaction capability, i.e., the elicitation module only understands simple answers, the dialogue manager must divide and simplify the users' inputs to make them understandable by the elicitation module. This is done by transforming the input into a set of communication and associated simple answer pairs. (The set may contain only one pair.) For example, the template [Entity1, Role1, new, Entity-Set2], corresponding to an input such as `the research group is presented by a new site` is divided into answering `yes` to the question "Is the research group presented by a site?" and then `new` to the question "Which site is presenting the research group?". This last question is created while processing the answer to the first one. The dialogue manager makes some assumptions as to which communications will be created when the first part of the new information input is processed. These assumptions are currently ad-hoc, i.e., they are coded in the dialogue manager. If the communications contained more information on what the dialogue manager could expect to happen next, these assumptions could be computed on-the-fly. How best to achieve this remains an open-issue (see section 11.2.3).

Because interpreting the new information is equivalent to several communications, the dialogue manager must also keep track of the focus evolution. This is done by calling the communication selection process with only one communication to select from (see section 5.3). This gives us the focus moves corresponding to the communication answered. Finally, any feedback communication created during the processing of the new information is immediately output, thus by-passing the communication selection process. This is done to ensure that users know exactly and without delay what changes the system has made in response to their input. We cannot rely on the focus rules here since they may delay the output of feedback which are necessary for the dialogue to continue (even though they may not be the most coherent ones). Outputting feedback is done the same way as outputting any presentation communication except that the communication collection process is modified to only collect feedback. The overall interpretation process follows therefore algorithm 6.2. Since the interpretation of new information requires some communications to already exist in the communication pool, not all input can be interpreted. For example, if the communication pool does not contain a communication about the creation of a site for the research group, the example presented above would not be processed. This way of interpreting inputs is quite similar to the one used in Collagen (see section 2.6.3) where users can select their input among a number of sentences proposed by the system. This process avoids the need for a highly efficient natural language understanding module which could be difficult to program (Allen 1987).

The templates interpreting new information are presented in table 6.10 with their asso-

Table 6.10 New information templates

Template	[Entity1, Role1, new, Entity-Set2]
Description	A new entity of type Entity-Set2 fills role Role1 in a new relation with existing entity Entity1
Action	• answer `yes` to ask-creation(?Relation1 | Entity1, Entity-Set2) • answer `new` to ask-role-filler(Relation1 | Role1, Entity-Set2)
Template	[Entity1, Role, Entity2]
Description	Existing entity Entity2 fills role Role in a new relation with existing entity Entity1
Action	• answer `yes` to ask-creation(?Relation1 | Entity1, Entity-Set2) • answer `Entity2` to ask-role-filler(Relation1 | Role, Entity-Set2)
Template	[Entity1, not, Role1, Entity-Set2]
Description	An entity of type Entity-Set2 does not fill role Role1 in a relation with existing entity Entity1
Action	• answer `no` to ask-creation(?Relation1 | Entity1, Entity-Set2)
Template	[Entity1, Attribute1, Value1]
Description	The value of attribute Attribute1 for entity Entity1 is Value1
Action	• answer `Value1` to ask-value(Entity1 | Attribute1)
Template	[Relation1, Attribute1, Value1]
Description	The value of attribute Attribute1 for relation Relation1 is Value1
Action	• answer `Value1` to ask-value(Relation1 | Attribute1)

ciated actions.

Example

Assume again that the model underlying the elicitation process is that of appendix A.1 and assume that the system is in the state shown in table 6.11 and that the last communication output was `ask-creation(p1 | rg1, site)`, i.e., "Do you want the research group to be presented by a site?". If the user answers `The research group is presented by a new page`, the template [Entity1, Role1, new, Entity-Set2] gets activated with Entity1 = #rg1, Role1 = `presented by` and Entity-Set2 = `page`. The first action of this template is to answer the question `ask-creation(?Relation1 | rg1, page)`. This answer makes the system assert relation #p2 and its associated roles. The system then removes relation #p1 and its associated roles from the specification as this relation can no longer be asserted without violating the model constraints. All communications related to #p1 are also removed from the communication pool. A presentation communication is added to the communication pool to present the changes made to the specification. Finally, the system outputs a question about the identity of the page presenting the research group. As a result, the system is in the state shown in table 6.12. The second action triggered by the template is to answer `new` to the question `ask-role-filler(p2 | presented by, page)`. The existence of page #pa1 is asserted and this new page fills relation `p2 presented by` role. A presentation communication is then added to the communication pool to present this change. Because this communication is more informative than the previous one about relation #p2, the previous communication is superseded (see section 5.2.1). The final state of the system is shown in table 6.13.

Table 6.11 Interpreting new information (system initial state)

Specification	Communication pool
entity(rg1, research group)	present(i1 | rs1, rg1)
entity(rs1, researcher set)	ask-creation(p1 | rg1, site)
? entity(?s1, site)	ask-creation(p2 | rg1, page)
? entity(?pa1, page)	**Focus set**
relation(i1, involve)	$\mathcal{F}_1 =_2 \{rg1, p1\}$
? relation(p1, present)	**Centres**
? relation(p2,present)	$Cf_2(rg1, p1)$
rolefillervalue(i1, involving, rs1)	$Cp_2(p1)$
rolefillervalue(i1, involved in, rg1)	$Cb_2(rg1)$
? rolefillervalue(p1, presenting, rg1)	
? rolefillervalue(p1, presented by, ?s1)	
? rolefillervalue(p2, presenting, rg1)	
? rolefillervalue(p2, presented by, ?pa1)	

Table 6.12 Interpreting new information (system intermediate state)

Specification	Communication pool
entity(rg1, research group)	present(i1 | rs1, rg1)
entity(rs1, researcher set)	present(p2 | rg1, page)
entity(?pa1, page)	ask-role-filler(p2 | presented by, page)
relation(i1, involve)	answer(ask-creation(p2 | rg1, page), yes)
relation(p2,present)	**Focus set**
rolefillervalue(i1, involving, rs1)	$\mathcal{F}_1 =_3 \{rg1, p2\}$
rolefillervalue(i1, involved in, rg1)	**Centres**
rolefillervalue(p2, presenting, rg1)	$Cf_3(rg1, p2)$
rolefillervalue(p2, presented by, ?pa1)	$Cp_3(p2)$
	$Cb_3(rg1)$

Table 6.13 Interpreting new information (system final state)

Specification	Communication pool
entity(rg1, research group)	present(i1 | rs1, rg1)
entity(rs1, researcher set)	present(p2 | rg1, pa1)
entity(pa1, page)	answer(ask-role-filler(p2 | presented by, page), new)
relation(i1, involve)	answer(ask-creation(p2 | rg1, page), yes)
relation(p2,present)	**Focus set**
rolefillervalue(i1, involving, rs1)	$\mathcal{F}_1 =_4 \{rg1, p2, pa1\}$
rolefillervalue(i1, involved in, rg1)	**Centres**
rolefillervalue(p2, presenting, rg1)	$Cf_4(p2, pa1)$
rolefillervalue(p2, presented by, pa1)	$Cp_4(pa1)$
	$Cb_4(p2)$

Table 6.14 Dialogue redirecting templates

Template	[speak, about, Entity1]
Description	Redirect the dialogue to Entity 1
Template	[speak, about, Relation1]
Description	Redirect the dialogue to Relation 1

6.4.3 Redirecting the dialogue

The next type of templates in the system is used to redirect the dialogue to a different topic. The redirection is done by outputting a presentation communication whose subjects include the new topic. This makes the focus evolve to the new topic and therefore redirects the dialogue to it. The system first tries to find a presentation communication where the new topic is the main subject. If there is no such communication, it then searches for a presentation communication where the new topic is a subject. If such a communication does not exist, the redirection action is ignored and a message informing the user of the problem is output. The presentation communication output is either an active communication in the communication pool or a communication that has already been presented. In that case, it is re-output. The selection of this communication is forced by calling the focus rules with only this communication to select from (see section 5.3). The templates redirecting the dialogue are presented in table 6.14.

A simplified form of dialogue redirection is allowed by the system for questions. This redirection consists of "passing" a question, i.e., refusing to answer it at the moment it is asked. In that case, the system simply removes the question from the communication pool and stores it in a special repository. Once all communications in the communication pool have been output, the content of this special repository is put back in the communication pool and the normal generation/interpretation process continues. The passing action is a special case of dialogue redirection since it asks the system to find another communication, but without specifying which one. It is useful for users who wants to avoid questions without taking the initiative. Waiting that all communications in the communication pool have been dealt with before re-considering the passed communications is simplistic. In some cases, it may be better to re-introduce them earlier. However, it is difficult to know what is the best time to re-output them, without a deep understanding of the structure of the dialogue and the reasons why users passed them.

Example

Assume again that the system is in the state shown in table 6.15 and that the last communication output was `ask-creation(p1 | rg1, site)`, i.e., "Do you want the research group to be presented by a site?". Now, if the user answers `I want to speak about the researcher set`, the system searches for a presentation communication related to researcher set #rs1. It first looks for a presentation communication where the researcher set is a main subject. Since there is no such communication, it then looks for a communication where the researcher set is a subject. Such a communication exists in the state of the dialogue considered here: present(i1 | rs1, rg1). It is therefore output. The system is then in the state shown in table 6.16 where we are expected to speak more about #rs1 ($Cp_3(i1)$).

Table 6.15 Dialogue redirecting (system initial state)

Specification	
entity(rg1, research group)	**Communication pool**
entity(rs1, researcher set)	present(i1 | rs1, rg1)
? entity(?s1, site)	ask-creation(p1 | rg1, site)
? entity(?pa1, page)	ask-creation(p2 | rg1, page)
relation(i1, involve)	**Focus set**
? relation(p1, present)	$\mathcal{F}_1 =_2 \{rg1, p1\}$
? relation(p2,present)	**Centres**
rolefillervalue(i1, involving, rs1)	$Cf_2(rg1, p1)$
rolefillervalue(i1, involved in, rg1)	$Cp_2(p1)$
? rolefillervalue(p1, presenting, rg1)	$Cb_2(rg1)$
? rolefillervalue(p1, presented by, ?s1)	
? rolefillervalue(p2, presenting, rg1)	
? rolefillervalue(p2, presented by, ?pa1)	

Table 6.16 Dialogue redirecting (system final state)

Specification	
entity(rg1, research group)	
entity(rs1, researcher set)	**Communication pool**
? entity(?s1, site)	ask-creation(p1 | rg1, site)
? entity(?pa1, page)	ask-creation(p2 | rg1, page)
relation(i1, involve)	**Focus set**
? relation(p1, present)	$\mathcal{F}_1 =_3 \{rg1, p1, i1, rs1\}$
? relation(p2,present)	**Centres**
rolefillervalue(i1, involving, rs1)	$Cf_3(i1, rs1, rg1)$
rolefillervalue(i1, involved in, rg1)	$Cp_3(i1)$
? rolefillervalue(p1, presenting, rg1)	$Cb_3(rg1)$
? rolefillervalue(p1, presented by, ?s1)	
? rolefillervalue(p2, presenting, rg1)	
? rolefillervalue(p2, presented by, ?pa1)	

6.4.4 Automatic answering

Enabling the system to answer some questions automatically is a way of providing users with less burdensome dialogues. We provide two templates for this purpose. These templates are interpreted by the dialogue manager. The dialogue manager will then look for questions to be answered automatically and answer them directly without outputting them. The users can therefore state general properties just once, while the dialogue manager ensures that all the individual relations are created. The templates interpreting general properties are presented in table 6.17 with the communications they answer automatically.

The way these templates operate is similar to the way templates interpreting new information work. The only difference is that the dialogue manager keeps the action part in memory and tries to apply them whenever it can, before the communications they answer are output (see section 5.6.1). For example, if the input "every new page is linked to home

Table 6.17 Automatic answering templates

Template	[every, new, Entity-Set, Role, Entity1]
Description	Entity1 (of type Entity-Set1) fills role Role in a new relation with each new instance of Entity-Set
Action	• answer `yes` to ask-creation(?Relation | New-Instance, Entity-Set1) • answer `Entity` to ask-role-filler(Relation | Role, Entity-Set1)
Template	[every, new, Entity-Set1, not, Role, Entity-Set2]
Description	Instances of Entity-Set2 do not fill role Role for any new instances of Entity-Set1
Action	• answer `no` to ask-creation(?Relation | New-Instance, Entity-Set2)

Table 6.18 Dialogue with user-initiative

S:	Research group #rg1 is the research group.
S:	Do you want a site presenting it ?
U:	Listen. (*Take the initiative*)
U:	The research group is presented by a new site.
S:	Site #s1 presents the research group.
U:	Site #s1 is described by a new home page.
S:	Home page #hp1 describes site #s1.
U:	I want to speak about the researcher set.
S:	OK. Researcher set #rs1 is the researcher set.
U:	#rs1 is presented by a new page.
S:	Page #pa1 presents the researcher set.
U:	Page #pa1 is linked to the home page.
S:	OK.
U:	OK. (*Release the initiative*)
S:	Home page #hp1 is linked from page #pa1. What is the title of page #pa1 ?
...	

page #hp" has been processed, the action of answering `yes` will be triggered each time a question about linking a new page to a home page is asked. Then the action of providing home page #hp as a role-filler for the link will be triggered.

6.4.5 Taking the initiative

In its mixed-initiative mode, the system asks questions and users answer them. Even when users input new information, the system keeps the initiative and continues to ask questions, though they should be related to the new information provided. In its user-initiative mode (see section 5.6.2), the system accepts new information (see section 6.4.2) and dialogue redirecting inputs (see section 6.4.3), but does not output questions until instructed to do so. This does not mean that the system does not generate questions. Questions *are* added to the communication pool since they are needed to interpret the user's inputs. They are simply not output by the dialogue manager. The system just provides feedback and alerts users when it does not understand the inputs. Since the system keeps track of the focus evolution when interpreting new information or redirecting the dialogue, it restarts asking questions where users stopped inputting their requirements. The mode of operating of the system is switched from mixed-initative to user-initiative and back by a special input (see section 6.4.6). An example of dialogue when the user takes the initiative is presented in table 6.18.

6.4.6 Special inputs

The system accepts some special inputs. These inputs are not considered part of the dialogue and therefore do not change the state of the system. These inputs include requests to see the current state of the specification, requests for help on how to use the system and requests to stop the system. Special inputs enable users to switch the initiative mode of the system by setting (`listen` input) or resetting (`OK` input) the `mixed-initiative mode` parameter (see section 5.6.2).

Chapter 7

Worked example

Objectives

The objectives of this chapter are to:

- Present an example of dialogue,
- Present the need for cooperation between the elicitation system and dialogue manager to manage the elicitation dialogue,
- Present the need for cooperation between the various theories used to manage the dialogue.

7.1 Algorithm

The process driving the generation and interpretation follows algorithm 7.1. It is the result of fleshing out the system general algorithm (see algorithm 4.1) with the more precise generation algorithms (see algorithms 5.2, 5.3 and 5.4) and interpretation algorithms (see algorithms 6.1 and 6.2). Actions where focus theories play a role are set in italics.

7.2 Example

The example we discuss in this chapter is presented in example 7.1. This dialogue is a slightly edited version of a real dialogue. The original version is presented in appendix B.1. The specification model corresponding to the example is that of appendix A.1. The theories used to generate outputs and interpret inputs are global focus theory (see section 3.2), local focus theory (see section 3.3) and present-first (see section 3.4), in order of decreasing importance.

7.3 Step-by-step generation and interpretation

In this section, we describe how each utterance of the system is generated and how each users' utterance is interpreted.

algorithm 7.1 Generating outputs and interpreting inputs

```
do in any order
  task 1 (Elicitation module)
    Produce communications into the Communication Pool
  end task 1
  task 2 (Elicitation module)
    Interpret simple answers from the Communication Pool
  end task 2
  task 3 (Dialogue manager)
    if the system is in mixed-initiative mode then
      repeat
        Collect active communications from communication pool
        Rank the communications
        Select best ranked communication
        Update focus state
        if an automatic answer can be provided then
          Answer the question directly
        else
          Select a generation template
          Generate anaphora
          Generate cue words
          Present the sentence to users
        end if
      until communication was not a presentation communication and was not answered
      automatically
    end if
    Display prompt for input
    Preprocess input
    Select an interpretation template
    if input is an answer then
      Add simple answer to the Communication Pool
    else if input is a new information then
      Divide input in simple answers based on the template
      repeat
        Find question communication related to a simple answer
        Update the focus state for this communication
        Add the simple answer to the Communication Pool
      until all simple answers have been processed
      Select and output any feedback
    else if input redirects the dialogue then
      Find a presentation communication related to the topic
      Select and output this communication
    else if input sets an automatic answer then
      Store new automatic answer
    else if input change the initiative then
      Set the initiative to "user" or "mixed" depending on the input
    end if
  end task 3
end do in any order
```

Example 7.1 Requirements elicitation dialogue

S1:	Research group #rg1 is the research group.
S2:	Do you want a site presenting it?
U1:	Yes.
S3:	A site presents the research group.
S4:	Which site presents the research group?
U2:	A new site.
S5:	Site #s1 presents the research group.
S6:	Do you want a home page describing site #s1?
U3:	Site #s1 is described by a new home page.
S7:	Home page #hp1 describes site #s1.
S8:	What is the title of home page #hp1?
U4:	Research group.
S9:	Do you want a page linked from it?
U5:	Yes.
S10:	A page is linked from research group home page #hp1.
S11:	Which page is linked from research group home page #hp1?
U6:	A new page.
S12:	Page #pa1 is linked from research group home page #hp1.
S13:	What is the title of page #pa1?
U7:	I would like to speak about the publication set
S14:	Publication set #ps1 is the publication set.
S15:	It is involved in the research group.
S16:	Do you want a site presenting it?

7.3.1 Output S1

The system starts by creating all the mandatory entities and relations. This depends on the minimum cardinalitites defined in the domain specific specification model used by the system (see appendix A). In our case, this involves the creations of a research group, a researcher set, a publication set and the `involve` relations between them. These creations are associated with the production of presentation communications describing the changes made to the specification. In particular, presentation communications of the entities asserted are created first. They are however replaced by presentation communications about the relations created since these communications are more informative (see section 5.2.1). Then, the system creates hypothetical entities and relations which represent possible changes to the specification. These creations are associated with question communications used to confirm or deny the proposed changes. The system state after these creations is presented in table 7.1. All these transformations are performed by the elicitation module as part of task 1 of algorithm 7.1. Given this initial state and the fact that no focus information is available to select the communication to output, the research group is presented by default. S1 is therefore produced. This triggers by default a no change global focus move and a no center local focus move. The system state becomes that of table 7.2.

Table 7.1 System initial state

Specification	
entity(rg1, research group)	rolefillervalue(i1, involving, rs1)
entity(rs1, researcher set)	rolefillervalue(i1, involved in, rg1)
entity(ps1, publication set)	rolefillervalue(i2, involving, ps1)
?entity(?s1, site)	rolefillervalue(i2, involved in, rg1)
?entity(?s2, site)	?rolefillervalue(p1, presenting, rg1)
?entity(?s3, site)	?rolefillervalue(p1, presented by, ?s1)
?entity(?pa1, page)	?rolefillervalue(p2, presenting, rg1)
?entity(?pa2, page)	?rolefillervalue(p2, presented by, ?pa1)
?entity(?pa3, page)	?rolefillervalue(p3, presenting, rs1)
relation(i1, involve)	?rolefillervalue(p3, presented by, ?s2)
relation(i2, involve)	?rolefillervalue(p4, presenting, rs1)
?relation(p1, present)	?rolefillervalue(p4, presented by, ?pa2)
?relation(p2, present)	?rolefillervalue(p5, presenting, ps1)
?relation(p3, present)	?rolefillervalue(p5, presented by, ?s3)
?relation(p4, present)	?rolefillervalue(p6, presenting, ps1)
?relation(p5, present)	?rolefillervalue(p6, presented by, ?pa3)
?relation(p6, present)	

Communication pool
present(rg1)
present(i1 | rs1, rg1)
present(i2 | ps1, rg1)
ask-creation(p1 | rg1, site)
ask-creation(p2 | rg1, page)
ask-creation(p3 | rs1, site)
ask-creation(p4 | rs1, page)
ask-creation(p5 | ps1, site)
ask-creation(p6 | ps1, page)

Focus set
$\mathcal{F}_1 =_0 \{\}$

Centres
$Cf_0(\emptyset)$
$Cp_0(\emptyset)$
$Cb_0(\emptyset)$

Table 7.2 System state (after S1)

Specification	Communication pool	Focus set
Same as in the initial state	present(i1 | rs1, rg1)	$\mathcal{F}_1 =_1 \{rg1\}$
	present(i2 | ps1, rg1)	**Centres**
	ask-creation(p1 | rg1, site)	$Cf_1(rg1)$
	ask-creation(p2 | rg1, page)	$Cp_1(rg1)$
	ask-creation(p3 | rs1, site)	$Cb_1(rg1)$
	ask-creation(p4 | rs1, page)	
	ask-creation(p5 | ps1, site)	
	ask-creation(p6 | ps1, page)	

7.3.2 Output S2

In order to choose the next communication to output, the normal selection process is applied, now that focus information is available (see task 3 in algorithm 7.1). The first theory to apply

Table 7.3 Communication ranks (for S2)

Communication	Global focus move $\mathcal{F}_1 =_1 \{rg1\}$	Local focus move $Cb_1(rg1), Cp_1(rg1), Cf_1(rg1)$
present(i1 | rs1, rg1)	additive	
present(i2 | ps1, rg1)	additive	
ask-creation(p1 | rg1, site)	no change $dir_1(\_, p1, rg1)$	retaining $Cb_1(\_, rg1), Cp_1(\_, p1)$
ask-creation(p2 | rg1, page)	no change $dir_1(\_, p2, rg1)$	retaining $Cb_1(\_, rg1), Cp_1(\_, p2)$
ask-creation(p3 | rs1, site)	digressing	
ask-creation(p4 | rs1, page)	digressing	
ask-creation(p5 | ps1, site)	digressing	
ask-creation(p6 | ps1, page)	digressing	

Table 7.4 System state (after S2)

Specification	Communication pool	Focus set
Same as after S1	present(i1 | rs1, rg1) present(i2 | ps1, rg1) ask-creation(p1 | rg1, site) ask-creation(p2 | rg1, page) ask-creation(p3 | rs1, site) ask-creation(p4 | rs1, page) ask-creation(p5 | ps1, site) ask-creation(p6 | ps1, page)	$\mathcal{F}_1 =_2 \{rg1, p1\}$ Centres $Cf_2(p1, rg1)$ $Cp_2(p1)$ $Cb_2(rg1)$

is the global focus theory since it is the more important (see section 5.3). The only communications associated with a no change global focus move are communications `ask-creation(p1 | rg1, site)` and `ask-creation(p2 | rg1, page)` (see figure 4.6). These two communications are therefore passed to the next formal theory, the local focus theory. On the other hand, the communications which are associated with either an additive or a digressing move are delayed, since a no change move is preferred over these moves. They will be output at a later stage when the state of the dialogue is suitable. The two communications passed to the local focus theory are both associated with retaining local focus moves. Therefore, the local focus theory cannot select one of them. The two communications are thus passed on to the present-first theory. Again, because the two communications are `ask-creation` communications, this theory cannot select one over the other. The ranking process therefore stops with the two communications on-par. This is summarised in table 7.3. (Global and focus spaces are repeated at the top of the table. Elements of no interest are replaced by underscores. The selected communication is set in bold face. The reasons for the focus moves of the selected communication are given in each focus move cell.) One communication is then selected at random by the selection process (see section 5.3). If we assume that communication `ask-creation(p1 | rg1, site)` is selected, the system state becomes that of table 7.4. Because research group #rg1 is the backward-looking centre after S2 was selected and the local focus move is a retaining move, it can be realised by a pronoun. On the other hand, since it is no longer the preferred centre of the communication, it is not put

Table 7.5 System state (after U1)

Specification	
entity(rg1, research group)	rolefillervalue(i1, involving, rs1)
entity(rs1, researcher set)	rolefillervalue(i1, involved in, rg1)
entity(ps1, publication set)	rolefillervalue(i2, involving, ps1)
entity(?s1, site)	rolefillervalue(i2, involved in, rg1)
?entity(?s2, site)	rolefillervalue(p1, presenting, rg1)
?entity(?s3, site)	rolefillervalue(p1, presented by, ?s1)
?entity(?pa2, page)	?rolefillervalue(p3, presenting, rs1)
?entity(?pa3, page)	?rolefillervalue(p3, presented by, ?s2)
relation(i1, involve)	?rolefillervalue(p4, presenting, rs1)
relation(i2, involve)	?rolefillervalue(p4, presented by, ?pa2)
relation(p1, present)	?rolefillervalue(p5, presenting, ps1)
?relation(p3, present)	?rolefillervalue(p5, presented by, ?s3)
?relation(p4, present)	?rolefillervalue(p6, presenting, ps1)
?relation(p5, present)	?rolefillervalue(p6, presented by, ?pa3)
?relation(p6, present)	

Communication pool
present(i1 | rs1, rg1)
present(i2 | ps1, rg1)
present(p1 | rg1, site)
ask-role-filler(p1 | presented by, site)
ask-creation(p3 | rs1, site)
ask-creation(p4 | rs1, page)
ask-creation(p5 | ps1, site)
ask-creation(p6 | ps1, page)

Focus set
$\mathcal{F}_1 =_2 \{rg1, p1\}$

Centres
$Cf_2(p1, rg1)$
$Cp_2(p1)$
$Cb_2(rg1)$

in subject position in the sentence. This position is filled by the site.

7.3.3 Input U1

Since the user answers yes to the question, `answer(ask-creation(p1 | rg1, site), yes)` is added to the communication pool. Then, the eliciation module interprets this answer and the specification is modified to reflect it (see task 2 of algorithm 7.1). In particular, relation #p1 and its associated roles are asserted in the specification; relation #p2 and its associated roles and communications are removed from the specification and the communication pool respectively; a new presentation communication to inform the user of the changes made and a new question communication about relation #p1's role-filler are added to the communication pool. The resulting system state is presented in table 7.5. (We do not represent the users' answers in the communication pool.)

7.3.4 Output S3

The communication selection process is run once more. The global focus theory is first applied. Only communications `present(p1 | rg1, site)` and `ask-role-filler(p1 |`

Table 7.6 Communication ranks (for S3)

Communication	Global focus move $\mathcal{F}_1 =_2 \{rg1, p1\}$	Local focus move $Cb_2(rg1), Cp_2(p1), Cf_2(p1, rg1)$
present(i1 | rs1, rg1)	additive	
present(i2 | ps1, rg1)	additive	
present(p1 | rg1, site)	no change $dir_2(\_, p1, p1)$	smooth-shift $Cb_2(\_, p1), Cp_2(\_, p1)$
ask-role-filler(p1 | presented by, site)	no change $dir_2(\_, p1, p1)$	smooth-shift $Cb_2(\_, p1), Cp_2(\_, p1)$
ask-creation(p3 | rs1, site)	digressing	
ask-creation(p4 | rs1, page)	digressing	
ask-creation(p5 | ps1, site)	digressing	
ask-creation(p6 | ps1, page)	digressing	

Table 7.7 System state (after S3)

Specification	Communication pool	Focus set
Same as after U1	present(i1 | rs1, rg1) present(i2 | ps1, rg1) ask-role-filler(p1 | presented by, site) ask-creation(p3 | rs1, site) ask-creation(p4 | rs1, page) ask-creation(p5 | ps1, site) ask-creation(p6 | ps1, page)	$\mathcal{F}_1 =_3 \{rg1, p1\}$ **Centres** $Cf_3(p1, rg1)$ $Cp_3(p1)$ $Cb_3(p1)$

`presented by, site)` are associated with a no change global focus move. They are therefore selected and passed on to the local focus theory. The two communications are associated with a smooth-shift local focus move. This is presented in table 7.6. They are therefore passed on to the present-first theory. Since communication `present(p1 | rg1, site)` is a presentation communication it is selected. The new system state is presented in table 7.7.

7.3.5 Output S4

The selection process is run again. The global focus theory is first applied. Only communication `ask-role-filler(p1 | presented by, site)` is associated with a no change global focus move. This is shown in table 7.8. This communication is therefore selected and output. The new system state is presented in 7.9. We can observe here a limitation of the natural language generator. Because relation `p1` is in focus and because relations are always output as full verbs, the generator is not able to benefit from the focus information (see section 5.4.1). A better generator could output a sentence such as "Which site is it?" where the focused information is left out.

7.3.6 Input U2

Input U2 is a simple answer. It is therefore asserted in the communication pool and directly interpreted by the elicitation module. The hypothetical site filling the `presented by` role

Table 7.8 Communication ranks (for S4)

Communication	Global focus move $\mathcal{F}_1 =_3 \{rg1, p1\}$	Local focus move $Cb_3(p1), Cp_3(p1), Cf_3(p1, rg1)$
present(i1 | rs1, rg1)	additive	
present(i2 | ps1, rg1)	additive	
ask-role-filler(p1 | presented by, site)	no change $dir_3(\_, p1, p1)$	continuation $Cb_3(\_, p1), Cp_3(\_, p1)$
ask-creation(p3 | rs1, site)	digressing	
ask-creation(p4 | rs1, page)	digressing	
ask-creation(p5 | ps1, site)	digressing	
ask-creation(p6 | ps1, page)	digressing	

Table 7.9 System state (after S4)

Specification	Communication pool	Focus set
Same as after S3	present(i1 | rs1, rg1) present(i2 | ps1, rg1) ask-role-filler(p1| presented by, site) ask-creation(p3 | rs1, site) ask-creation(p4 | rs1, page) ask-creation(p5 | ps1, site) ask-creation(p6 | ps1, page)	$\mathcal{F}_1 =_4 \{rg1, p1\}$ Centres $Cf_4(p1)$ $Cp_4(p1)$ $Cb_4(p1)$

Table 7.10 System state (after U2)

Specification	
entity(rg1, research group)	rolefillervalue(i1, involving, rs1)
entity(rs1, researcher set)	rolefillervalue(i1, involved in, rg1)
entity(ps1, publication set)	rolefillervalue(i2, involving, ps1)
entity(s1, site)	rolefillervalue(i2, involved in, rg1)
?entity(?hp1, home page)	rolefillervalue(p1, presenting, rg1)
?entity(?s2, site)	rolefillervalue(p1, presented by, s1)
?entity(?s3, site)	?rolefillervalue(d1, describing, s1)
?entity(?pa2, page)	?rolefillervalue(d1, described by, ?hp1)
?entity(?pa3, page)	?rolefillervalue(p3, presenting, rs1)
relation(i1, involve)	?rolefillervalue(p3, presented by, ?s2)
relation(i2, involve)	?rolefillervalue(p4, presenting, rs1)
relation(p1, present)	?rolefillervalue(p4, presented by, ?pa2)
?relation(d1, describe)	?rolefillervalue(p5, presenting, ps1)
?relation(p3, present)	?rolefillervalue(p5, presented by, ?s3)
?relation(p4, present)	?rolefillervalue(p6, presenting, ps1)
?relation(p5, present)	?rolefillervalue(p6, presented by, ?pa3)
?relation(p6, present)	

Communication pool	Focus set
present(i1 | rs1, rg1)	$\mathcal{F}_1 =_4 \{rg1, p1\}$
present(i2 | ps1, rg1)	
present(p1 | rg1, s1)	**Centres**
ask-creation(d1 | s1, home page)	$Cf_4(p1)$
ask-creation(p3 | rs1, site)	$Cp_4(p1)$
ask-creation(p4 | rs1, page)	$Cb_4(p1)$
ask-creation(p5 | ps1, site)	
ask-creation(p6 | ps1, page)	

Table 7.11 Communication ranks (for S5)

Communication	Global focus move $\mathcal{F}_1 =_4 \{rg1, p1\}$	Local focus move $Cb_4(p1), Cp_4(p1), Cf_4(p1)$
present(i1 | rs1, rg1)	additive	
present(i2 | ps1, rg1)	additive	
present(p1 | rg1, s1)	no change $dir_4(\_, p1, p1)$	continuation $Cb_4(\_, p1), Cp_4(\_, p1)$
ask-creation(d1 | s1, home page)	digressing	
ask-creation(p3 | rs1, site)	digressing	
ask-creation(p4 | rs1, page)	digressing	
ask-creation(p5 | ps1, site)	digressing	
ask-creation(p6 | ps1, page)	digressing	

Table 7.12 System state (after S5)

Specification	Communication pool	Focus set
Same as after U2	present(i1 | rs1, rg1) present(i2 | ps1, rg1) ask-creation(d1 | s1, home page) ask-creation(p3 | rs1, site) ask-creation(p4 | rs1, page) ask-creation(p5 | ps1, site) ask-creation(p6 | ps1, page)	$\mathcal{F}_1 =_5 \{rg1, p1, s1\}$ **Centres** $Cf_5(p1, rg1, s1)$ $Cp_5(p1)$ $Cb_5(p1)$

of relation #p1 is asserted in the specification. A presentation communication presenting this change is added to the communication pool. The change also leads to the creation of hypothetical elements related to this new site and to the creation of new question communications. The new system state after the interpretation of input U2 is presented in table 7.10.

7.3.7 Output S5

Only communication `present(p1 | rg1, s1)` is associated with a no change global focus move as shown in table 7.11. Communication `present(p1 | rg1, s1)` is therefore selected and output. The new system state after S5 is presented in table 7.12.

7.3.8 Output S6

Only communication `ask-creation(d1 | s1, home page)` is now associated with a no change global focus move as shown in table 7.13. It is therefore immediately selected and output. The system state becomes that of table 7.14.

7.3.9 Input U3 and output S7

Input U3 is an input providing new information. It corresponds to answering `yes` to "Do you want a home page describing site #s1?" and then `new` to "Which home page describes

Table 7.13 Communication ranks (for S6)

Communication	Global focus move $\mathcal{F}_1 =_5 \{rg1, p1, s1\}$	Local focus move $Cb_5(p1), Cp_5(p1), Cf_5(p1)$
present(i1 | rs1, rg1)	additive	
present(i2 | ps1, rg1)	additive	
ask-creation(d1 | s1, home page)	no change $dir_5(\_, d1, s1)$	rough-shift $Cb_5(\_, s1), Cp_5(\_, d1)$
ask-creation(p3 | rs1, site)	digressing	
ask-creation(p4 | rs1, page)	digressing	
ask-creation(p5 | ps1, site)	digressing	
ask-creation(p6 | ps1, page)	digressing	

Table 7.14 System state (after S6)

Specification	Communication pool	Focus set
Same as after S5	present(i1 | rs1, rg1) present(i2 | ps1, rg1) ask-creation(d1 | s1, home page) ask-creation(p3 | rs1, site) ask-creation(p4 | rs1, page) ask-creation(p5 | ps1, site) ask-creation(p6 | ps1, page)	$\mathcal{F}_1 =_6 \{rg1, p1, s1\}$ Centres $Cf_6(d1, s1)$ $Cp_6(d1)$ $Cb_6(s1)$

Table 7.15 System state (after S7)

Specification	
entity(rg1, research group)	rolefillervalue(i1, involving, rs1)
entity(rs1, researcher set)	rolefillervalue(i1, involved in, rg1)
entity(ps1, publication set)	rolefillervalue(i2, involving, ps1)
entity(s1, site)	rolefillervalue(i2, involved in, rg1)
entity(hp1, home page)	rolefillervalue(p1, presenting, rg1)
?entity(?pa1, page)	rolefillervalue(p1, presented by, s1)
?entity(?hp2, home page)	rolefillervalue(d1, describing, s1)
?entity(?s2, site)	rolefillervalue(d1, described by, hp1)
?entity(?s3, site)	?rolefillervalue(l1, linked from, hp1)
?entity(?pa2, page)	?rolefillervalue(l1, linked to, ?pa1)
?entity(?pa3, page)	?rolefillervalue(l2, linked from, hp1)
relation(i1, involve)	?rolefillervalue(l2, linked to, ?hp2)
relation(i2, involve)	?rolefillervalue(p3, presenting, rs1)
relation(p1, present)	?rolefillervalue(p3, presented by, ?s2)
relation(d1, describe)	?rolefillervalue(p4, presenting, rs1)
?relation(l1, link)	?rolefillervalue(p4, presented by, ?pa2)
?relation(l2, link)	?rolefillervalue(p5, presenting, ps1)
?relation(p3, present)	?rolefillervalue(p5, presented by, ?s3)
?relation(p4, present)	?rolefillervalue(p6, presenting, ps1)
?relation(p5, present)	?rolefillervalue(p6, presented by, ?pa3)
?relation(p6, present)	

Communication pool
present(i1 | rs1, rg1)
present(i2 | ps1, rg1)
ask-value(hp1 | title)
ask-creation(l1 | hp1, page)
ask-creation(l2 | hp1, home page)
ask-creation(p3 | rs1, site)
ask-creation(p4 | rs1, page)
ask-creation(p5 | ps1, site)
ask-creation(p6 | ps1, page)

Focus set
$\mathcal{F}_1 =_7 \{rg1, p1, s1, d1, hp1\}$
Centres
$Cf_7(d1, s1, hp1)$
$Cp_7(d1)$
$Cb_7(d1)$

Table 7.16 Communication ranks (for S8)

Communication	Global focus move $\mathcal{F}_1 =_7 \{rg1, p1, s1, d1, hp1\}$	Local focus move $Cb_7(d1), Cp_7(d1)$ $Cf_7(d1, s1, hp1)$
present(i1 | rs1, rg1)	additive	
present(i2 | ps1, rg1)	additive	
ask-value(hp1 | title)	no change $dir_7(\_, hp1, hp1)$	smooth-shift $Cb_7(\_, hp1), Cp_7(\_, hp1)$
ask-creation(l1 | hp1, page)	additive	
ask-creation(l2 | hp1, home page)	additive	
ask-creation(p3 | rs1, site)	digressing	
ask-creation(p4 | rs1, page)	digressing	
ask-creation(p5 | ps1, site)	digressing	
ask-creation(p6 | ps1, page)	digressing	

site #s1". The communication pool and then the specification is therefore modified to reflect these answers. Finally, the presentation communication `present(d1 | s1, hp1)` presenting the changes made in the specification is presented as a feedback to the user. The resulting system state is presented in table 7.15. The selection process is by-passed during the processing of input U3 and output S7 since we want to provide immediate feedback to the user, and not let the dialogue manager decide when to output them. However, the focus evolution is tracked so that the system restarts outputting communications in the right context.

7.3.10 Output S8 and input U4

From the active communication collected in the communication pool only communication `ask-value(hp1 | title)` is associated with a no change global focus move. This is described in table 7.16. Therefore, this communication is immediately selected and output. The answer of the user is a simple answer and can be processed directly by the elicitation module once it has been asserted in the communication pool. The resulting system state is presented in table 7.17.

7.3.11 Output S9

The selection process is run again. Communications `present(i1 | rs1, rg1)`, `present(i2 | ps1, rg1)`, `ask-creation(l1 | hp1, page)` and `ask-creation( l2 | hp1, home page)` are all associated with an additive global focus move. There is no communication associated with a no change global focus move. These communications are therefore passed on to the local focus theory. Communications `ask-creation(l1 | hp1, page)` and `ask-creation(l2 | hp1, home page)` are associated with a retaining local focus move. Communications `present(i1 | rs1, rg1)` and `present(i2 | ps1, rg1)` are associated with a no centre local focus move. This is shown in table 7.18. Communications `ask-creation(l1 | hp1, page)` and `ask-creation(l2 | hp1, home page)` are then passed on to the present-first theory. Because the two communications are question communications, this theory does not make any difference between them. One of them is therefore

Table 7.17 System state (after U4)

Specification	
entity(rg1, research group)	rolefillervalue(i1, involving, rs1)
entity(rs1, researcher set)	rolefillervalue(i1, involved in, rg1)
entity(ps1, publication set)	rolefillervalue(i2, involving, ps1)
entity(s1, site)	rolefillervalue(i2, involved in, rg1)
entity(hp1, home page)	rolefillervalue(p1, presenting, rg1)
?entity(?pa1, page)	rolefillervalue(p1, presented by, s1)
?entity(?hp2, home page)	rolefillervalue(d1, describing, s1)
?entity(?s2, site)	rolefillervalue(d1, described by, hp1)
?entity(?s3, site)	?rolefillervalue(l1, linked from, hp1)
?entity(?pa2, page)	?rolefillervalue(l1, linked to, ?pa1)
?entity(?pa3, page)	?rolefillervalue(l2, linked from, hp1)
relation(i1, involve)	?rolefillervalue(l2, linked to, ?hp2)
relation(i2, involve)	?rolefillervalue(p3, presenting, rs1)
relation(p1, present)	?rolefillervalue(p3, presented by, ?s2)
relation(d1, describe)	?rolefillervalue(p4, presenting, rs1)
?relation(l1, link)	?rolefillervalue(p4, presented by, ?pa2)
?relation(l2, link)	?rolefillervalue(p5, presenting, ps1)
?relation(p3, present)	?rolefillervalue(p5, presented by, ?s3)
?relation(p4, present)	?rolefillervalue(p6, presenting, ps1)
?relation(p5, present)	?rolefillervalue(p6, presented by, ?pa3)
?relation(p6, present)	attributevalue(hp1, title, "Research group")

Communication pool
present(i1 | rs1, rg1)
present(i2 | ps1, rg1)
ask-creation(l1 | hp1, page)
ask-creation(l2 | hp1, home page)
ask-creation(p3 | rs1, site)
ask-creation(p4 | rs1, page)
ask-creation(p5 | ps1, site)
ask-creation(p6 | ps1, page)

Focus set
$\mathcal{F}_1 =_8 \{rg1, p1, s1, d1, hp1\}$

Centres
$Cf_8(hp1)$
$Cp_8(hp1)$
$Cb_8(hp1)$

Table 7.18 Communication ranks (for S9)

Communication	Global focus move	Local focus move
	$\mathcal{F}_1 =_8 \{rg1, p1, s1, d1, hp1\}$	$Cb_8(hp1), Cp_8(hp1)$ $Cf_8(hp1)$
present(i1 | rs1, rg1)	additive $spec_8(\_, i1, rg1)$	no centre $Cb_8(\_, \emptyset), Cp_8(\_, i1)$
present(i2 | ps1, rg1)	additive $spec_8(\_, i2, rg1)$	no centre $Cb_8(\_, \emptyset), Cp_8(\_, i2)$
ask-creation(l1 | hp1, page)	additive $spec_8(\_, l1, hp1)$	retaining $Cb_8(\_, hp1), Cp_8(\_, l1)$
ask-creation(l2 | hp1, home page	additive $spec_8(\_, l2, hp1)$	retaining $Cb_8(\_, hp1), Cp_8(\_, l2)$
ask-creation(p3 | rs1, site)	digressing	
ask-creation(p4 | rs1, page)	digressing	
ask-creation(p5 | ps1, site)	digressing	
ask-creation(p6 | ps1, page)	digressing	

Table 7.19 System state (after S9)

Specification	Communication pool	Focus set
Same as after U4	present(i1 | rs1, rg1) present(i2 | ps1, rg1) ask-creation(l1 | hp1, page) ask-creation(l2 | hp1, home page) ask-creation(p3 | rs1, site) ask-creation(p4 | rs1, page) ask-creation(p5 | ps1, site) ask-creation(p6 | ps1, page)	$\mathcal{F}_1 =_9 \{rg1, p1, s1, d1, hp1\}$ $\mathcal{F}_2 =_9 \{l1\}$ controlling$_9(\mathcal{F}_1, \mathcal{F}_2)$ **Centres**: $Cf_9(l1, hp1)$ $Cp_9(l1)$ $Cb_9(hp1)$

Table 7.20 System state (after U6)

Specification	
entity(rg1, research group)	rolefillervalue(i1, involving, rs1)
entity(rs1, researcher set)	rolefillervalue(i1, involved in, rg1)
entity(ps1, publication set)	rolefillervalue(i2, involving, ps1)
entity(s1, site)	rolefillervalue(i2, involved in, rg1)
entity(hp1, home page)	rolefillervalue(p1, presenting, rg1)
entity(pa1, page)	rolefillervalue(p1, presented by, s1)
?entity(?hp2, home page)	rolefillervalue(d1, describing, s1)
?entity(?s2, site)	rolefillervalue(d1, described by, hp1)
?entity(?s3, site)	rolefillervalue(l1, linked from, hp1)
?entity(?pa2, page)	rolefillervalue(l1, linked to, pa1)
?entity(?pa3, page)	?rolefillervalue(l2, linked from, hp1)
relation(i1, involve)	?rolefillervalue(l2, linked to, ?hp2)
relation(i2, involve)	?rolefillervalue(p3, presenting, rs1)
relation(p1, present)	?rolefillervalue(p3, presented by, ?s2)
relation(d1, describe)	?rolefillervalue(p4, presenting, rs1)
relation(l1, link)	?rolefillervalue(p4, presented by, ?pa2)
?relation(l2, link)	?rolefillervalue(p5, presenting, ps1)
?relation(p3, present)	?rolefillervalue(p5, presented by, ?s3)
?relation(p4, present)	?rolefillervalue(p6, presenting, ps1)
?relation(p5, present)	?rolefillervalue(p6, presented by, ?pa3)
?relation(p6, present)	attributevalue(hp1, title, "Research group")

<table>
<tr><th>Communication pool</th><th>Focus set</th></tr>
<tr><td>present(i1 | rs1, rg1)
present(i2 | ps1, rg1)
present(l1 | hp1, pa1)
ask-creation(l2 | hp1, home page)
ask-creation(p3 | rs1, site)
ask-creation(p4 | rs1, page)
ask-creation(p5 | ps1, site)
ask-creation(p6 | ps1, page)</td><td>$\mathcal{F}_1 =_{l1} \{rg1, p1, s1, d1, hp1\}$
$\mathcal{F}_2 =_{l1} \{l1\}$
controlling$_{l1}(\mathcal{F}_1, \mathcal{F}_2)$
Centres
$Cf_{l1}(l1, hp1)$
$Cp_{l1}(l1)$
$Cb_{l1}(hp1)$</td></tr>
</table>

selected at random. If we assume that communication `ask-creation(l1 | hp1, page)` is selected, S9 is output. A new focus space is created by the action of the additive global focus move. It will contain everything related to the new page. The new system state is presented in table 7.19. Because the home page is the backward-looking centre of the current sentence and was the backward-looking centre of the previous sentence too, it is realised by a pronoun.

7.3.12 Input U5 to input U6

What happens in the rest of the example until input U6 is equivalent to what happened between input U1 and input U2. The main difference is that all the new entities spoken about are put in the new global focus space $\mathcal{F}_2$. The state of the system after U6 is presented in table 7.20. Note that after U5 the title of the home page is used in the generation process. This makes it easier to recognise the home page spoken about since its name is more explicit than a simple identifier.

7.3.13 Outputs S12 and S13

What happens for outputs S12 and S13 is equivalent to what happened for outputs S7 and S8.

7.3.14 Input U7 and output S14

Input U7 is a dialogue redirecting input. Since the user wants to discuss the publication set #ps1, the system searches for a communication related to it. The presentation communication most directly related to it is `present(ps1)`. This communication was created at the start of the dialogue but was immediately dismissed because of the more informative communication `present(i2 | ps1, rg1)` (see section 7.3.1). This communication is therefore renewed, selected and output (see section 6.4.3). The global focus move associated with this relation is a pop/additive move. Focus space $\mathcal{F}_2$ is therefore closed and a new focus space is created. This move is also associated with a no centre local focus move. The system state becomes that of table 7.21.

7.3.15 Outputs S15 and S16

The selection process restarts in the state described in table 7.21. What happens is then equivalent to what happened during the selection of S1 and S2, except that the role of research group #rg1 is played by publication set #ps1 in the new focus space and that of relations #p1 and #p2 are played by relations #p5 and #p6 respectively. The state of the system after S16 is presented in table 7.22.

7.4 Comments

There are two main points that appear from the example presented above. The first point is the complementarity of the elicitation module and dialogue manager. The way the system works is defined conjointly by these two processes. The dialogue manager is restricted in its choices by what is in the communication pool. However, it restricts what the elicitation module can do by selecting which communication will be output. Users still keep the ultimate control over the process since they can redirect the dialogue and therefore can decide what the elicitation module is working on. The second point is the complementarity of the focus theories. The global focus theories defines the overall structure of the dialogue grouping the elements spoken about in a few spaces: $\mathcal{F}_1$ for everything related to the presentation of the research group, $\mathcal{F}_2$ for everything related to the new page, and $\mathcal{F}_3$ for everything related to the presentation of the publication set. This is shown in figure 7.1. (Some entities are represented more than once as they belong to several spaces. In $\mathcal{F}_2$, the dialogue is first about relation #i2 and research group #rg1 but then shifts to relation #p5.) By defining which space is active, the theory orients the dialogue and the elicitation module in an orderly fashion. The local theory has then two roles. The first role is to organise the presentation of the information belonging to a focus space. For example the theory ensures the smooth

Table 7.21 System state (after S14)

Specification	
entity(rg1, research group)	rolefillervalue(i1, involving, rs1)
entity(rs1, researcher set)	rolefillervalue(i1, involved in, rg1)
entity(ps1, publication set)	rolefillervalue(i2, involving, ps1)
entity(s1, site)	rolefillervalue(i2, involved in, rg1)
entity(hp1, home page)	rolefillervalue(p1, presenting, rg1)
entity(pa1, page)	rolefillervalue(p1, presented by, s1)
?entity(?hp2, home page)	rolefillervalue(d1, describing, s1)
?entity(?s2, site)	rolefillervalue(d1, described by, hp1)
?entity(?s3, site)	rolefillervalue(l1, linked from, hp1)
?entity(?pa2, page)	rolefillervalue(l1, linked to, pa1)
?entity(?pa3, page)	?rolefillervalue(l2, linked from, hp1)
relation(i1, involve)	?rolefillervalue(l2, linked to, ?hp2)
relation(i2, involve)	?rolefillervalue(p3, presenting, rs1)
relation(p1, present)	?rolefillervalue(p3, presented by, ?s2)
relation(d1, describe)	?rolefillervalue(p4, presenting, rs1)
relation(l1, link)	?rolefillervalue(p4, presented by, ?pa2)
?relation(l2, link)	?rolefillervalue(p5, presenting, ps1)
?relation(p3, present)	?rolefillervalue(p5, presented by, ?s3)
?relation(p4, present)	?rolefillervalue(p6, presenting, ps1)
?relation(p5, present)	?rolefillervalue(p6, presented by, ?pa3)
?relation(p6, present)	attributevalue(hp1, title, "Research group")

Communication pool	Focus set	Centres
present(i1 | rs1, rg1)	$\mathcal{F}_1 =_{14} \{rg1, p1, s1, d1, hp1\}$	$Cf_{14}(i2, ps1, rg1)$
ask-creation(l2 | hp1, home page)	$\mathcal{F}_2 =_{14} \{l1, pa1, hp1\}$	$Cp_{14}(i2)$
ask-creation(p3 | rs1, site)	$\mathcal{F}_3 =_{14} \{i2, ps1, rg1\}$	$Cb_{14}(i2)$
ask-creation(p4 | rs1, page)	closed $_{14}(\mathcal{F}_2)$	
ask-creation(p5 | ps1, site)	controlling $_{1}4(\mathcal{F}_1, \mathcal{F}_3)$	
ask-creation(p6 | ps1, page)		

Table 7.22 System state (after S16)

Specification	Communication pool	Focus set
Same as after S14	ask-creation(l2 | hp1, home page)	$\mathcal{F}_1 =_{16} \{rg1, p1, s1, d1, hp1\}$
	ask-creation(p3 | rs1, site)	$\mathcal{F}_2 =_{16} \{l1, pa1, hp1\}$
	ask-creation(p4 | rs1, page)	$\mathcal{F}_3 =_{16} \{i2, ps1, rg1, p5\}$
	ask-creation(p5 | ps1, site)	closed $_{16}(\mathcal{F}_2)$
	ask-creation(p6 | ps1, page)	controlling $_{16}(\mathcal{F}_1, \mathcal{F}_3)$
		Centres
		$Cf_{16}(p5, ps1)$
		$Cp_{16}(ps1)$
		$Cb_{16}(ps1)$

going from research group to site to home page. The second role of the local theory is to help choose which focus space should next become active when several choices are possible. For example, the theory places a preference on creating $\mathcal{F}_2$ before $\mathcal{F}_3$ because, at the time the creation was made (sentence S9), $\mathcal{F}_2$ was more related to the current focus than $\mathcal{F}_3$ was.

A summary of the focus moves made during the dialogue is given in table 7.23. We can note that local continution moves are associated with global no change moves and global pop moves with local no centre move as discussed in section 2.4.1.

Figure 7.1 Dialogue evolution

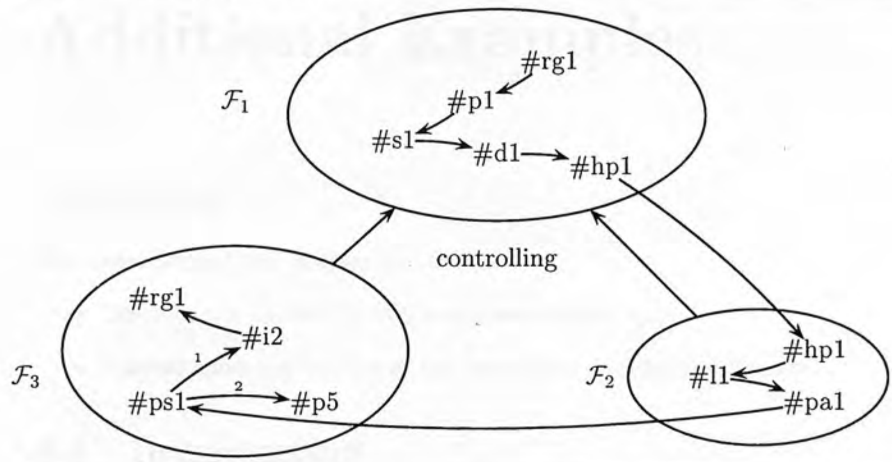


Table 7.23 Focus evolution during the example dialogue

		Global focus moves	Local focus moves			Global focus moves	Local focus moves
Communications	S1	no change	no centre	Communications	U4	—	—
	S2	no change	retaining		S9	additive	retaining
	U1	—	—		U5	—	—
	S3	no change	smooth shift		S10	no change	smooth shift
	S4	no change	continuation		S11	no change	continuation
	U2	—	—		U6	—	—
	S5	no change	continuation		S12	no change	continuation
	S6	no change	rough shift		S13	no change	smooth shift
	U3	no change	smooth shift		U7	—	—
		no change	continuation		S14	pop/additive	no centre
	S7	no change	continuation		S15	no change	retaining
	S8	no change	smooth shift		S16	no change	retaining

Chapter 8

Additional Examples

Objectives

The objectives of this chapter are to:

- Describe the capability of the system to scale up,
- Present some limitations of our treatment of natural language.

8.1 Introduction

An important issue when developing systems for requirements elicitation is how well they scale up. In the previous chapters we have presented the operating of our system using a small specification model. In this chapter, we use other models in order to appreciate how our system deals with them. In particular, we investigate three topics:

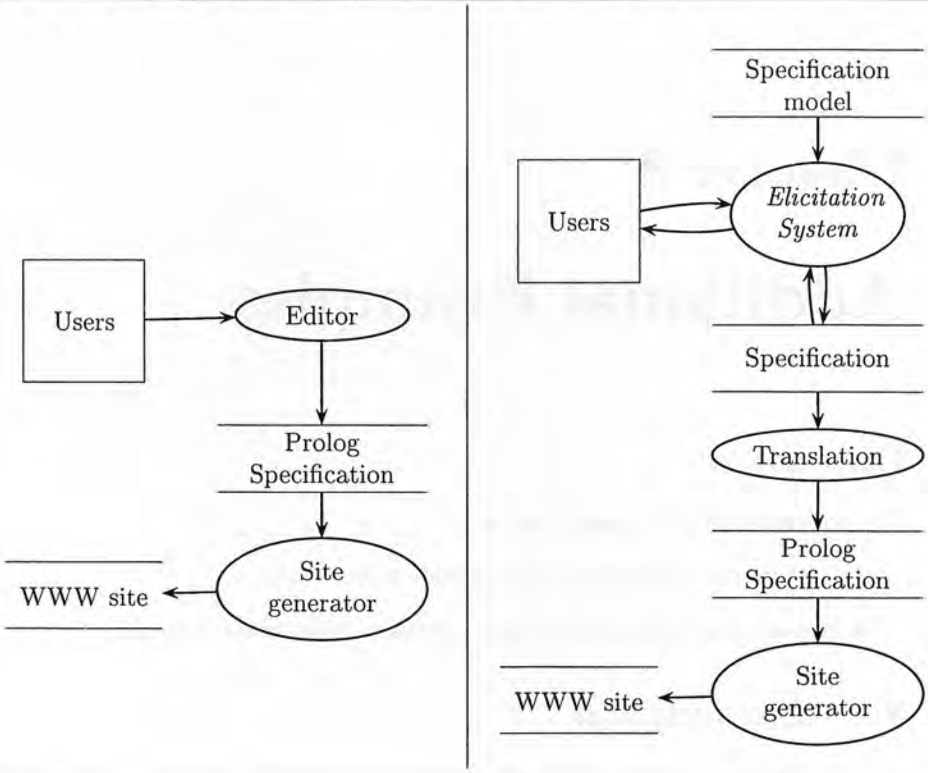
1. how does our system behave when the specification models become big enough to deal with real problems?
2. how does our system behave with existing entity-relationship models and what are the limitations imposed on the models to be usable by the system?
3. can the domain specific specification model (see section 4.2) be acquired in the same way as requirements?

Although this chapter only provides hints at the capabilities of our system to scale up and at the problems in doing so, it still provides a way of appreciating them. A precise evaluation of the role of the focus theories on the quality of the elicitation dialogue is given in chapter 9.

8.2 Full WWW Site

The current WWW site of the Software Systems and Processes Research Group of the University of Edinburgh (http://www.dai.ed.ac.uk/groups/ssp/index.html) is automatically built from a specification in Prolog. Rather than having to modify HTML code, only the

Figure 8.1 WWW site generation (without and with our system)



specification needs to be changed. The changes are then propagated to all the relevant pages when the site is re-generated. This process is shown on the left hand side of figure 8.1. A problem with this approach is the need to understand the Prolog specification representing the WWW site design. Using our system to hide these details seems a good idea. Therefore, we replaced the simple editor by our elicitation system as shown on the right hand side of figure 8.1. The model on which the specification is based is presented in appendix A.2. The specification produced by our system is then translated into the Prolog specification used by the site generator. Because the specification model used by our system is slightly more complete than the initial model which does not contain any information on the way pages should be presented (this is compiled out in the site generator), we have adapted the site generator to take into account some of the new information available. An extract of a dialogue with the system is given in example 8.1. The corresponding part of the specification is shown in table 8.1. This part of the specification is then translated to a Prolog term like the following: `event(Monday, 1998, 10, 'F13', 'Researcher1', talk1, talk1_abstract.txt, talk)`. (The type of the event, i.e., talk, is not presented in the example dialogue nor in the part of specification shown here.) Finally, this term is processed by the site generator. The resulting page in the WWW site is presented in figure 8.2 with the corresponding page from the real site. The conclusions we can draw from this small experiment are:

Example 8.1 Full World-Wide-Web site elicitation dialogue

S: Research group #rg1 is a research group.
What is its name?
U: "Software Systems and Processes"
S: What is its mail address?
U: I want to speak about the event set.
S: OK. Event set #es1 is a event set.
It is involved by research group Software Systems and Processes.
Do you want an event belonging to event set #es1?
U: yes.
S: An event belongs to event set #es1.
Which event belongs to event set #es1?
U: A new one.
...
S: Event #e1 belongs to event set #es1.
What is the title of event #e1?
U: talk1
S: What is its time?
U: 10
S: What is its day?
U: Monday
S: What is its year?
U: 1998
S: What is its location?
U: F13
S: What is its abstract?
U: talk1_abstract.txt
S: A researcher presents it.
Which researcher presents event #e1?
U: A new researcher.
S: researcher #r1 presents event #e1.
U: What is the name of researcher #r1?
S: Researcher1
...

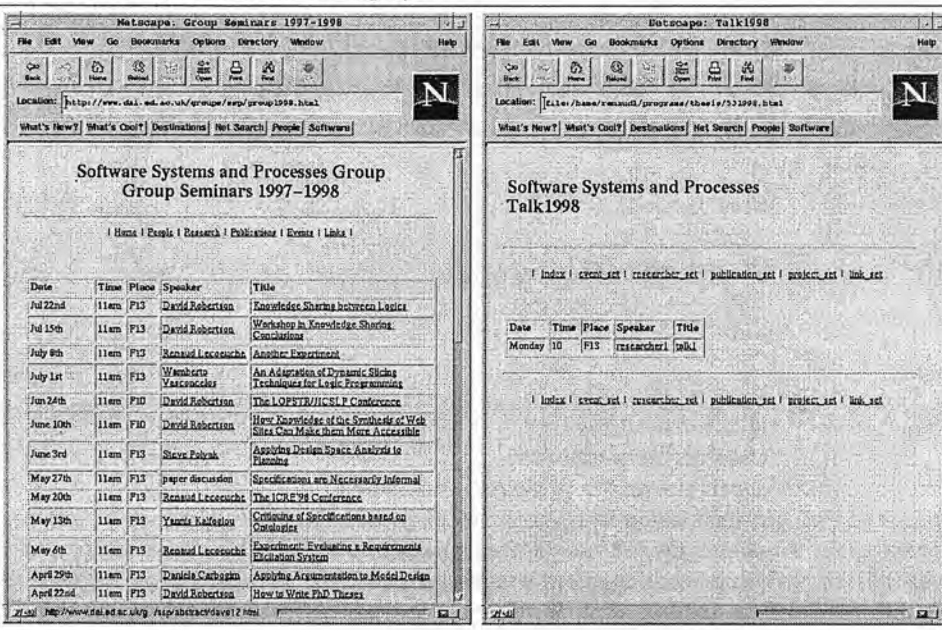
- Extending the specification model to cover a full WWW site is possible and translating the specification for use by another program has been easy.

- The dialogue resulting from the extension is not easy to understand however. Because numerous entities and relations are introduced, the dialogue requires a lot of attention. Focus rules help simplify the dialogue by constraining it to a few entities at a time, but more "radical" techniques such as graphics may be needed to deal with big specifications. Focus theories that do not rely on natural language phenomena may still be used in graphical interfaces to direct the system to what users are paying attention to.

Table 8.1 Specification of an event

Specification	
entity(e1, event)	rolefiller(b1, belonging to, es1)
entity(es1, event set)	rolefiller(b1, containing, e1)
entity(r1, researcher)	rolefiller(p1, presenting, e1)
relation(b1, belong es)	rolefiller(p1, presented by, r1)
relation(p1, present)	attributevalue(e1, title, talk1)
attributevalue(e1, time, 10)	attributevalue(e1, day, Monday)
attributevalue(e1, year, 1998)	attributevalue(e1, location, F13)
attributevalue(e1, abstract, talk1_abstract.txt)	attributevalue(r1, name, Researcher1)

Figure 8.2 WWW page example (the page from the real site is on the left and the page created with our system is on the right)



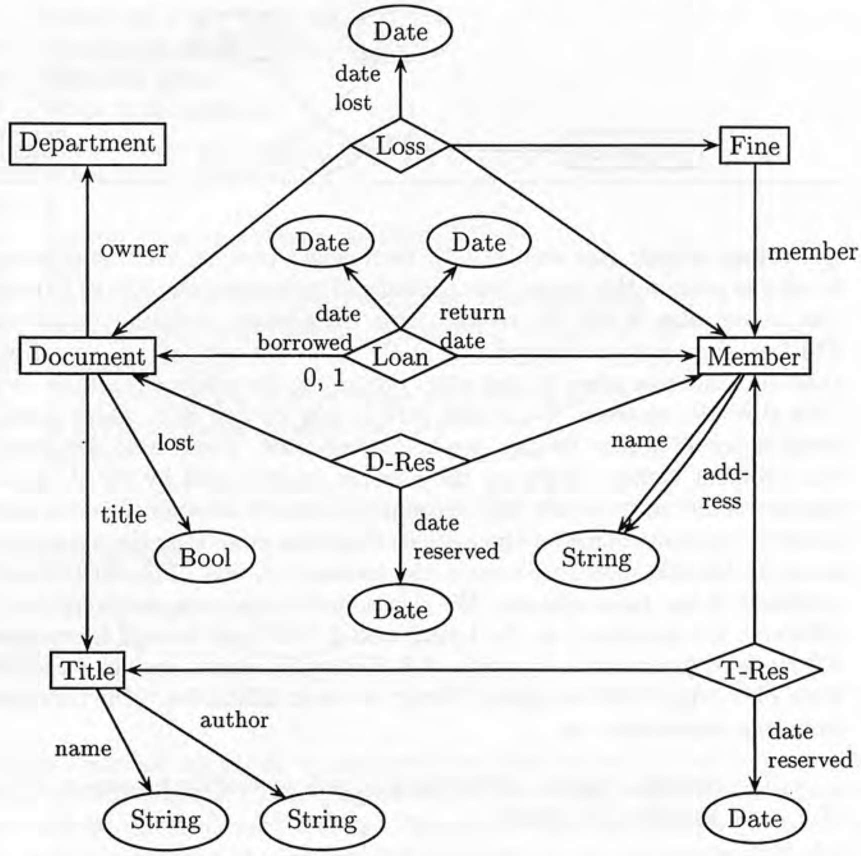
- Given the previous remark, our system may be better suited to write small specifications. When dealing with big specifications, the system could be used to help make changes to the specification. By focusing on a small part and ensuring the correctness and completeness of the specification, the system seems well suited for this task.

8.3 Another domain

All the examples presented in this thesis, except for the preceding section, have been based on the specification model of appendix A.1. This poses two problems:

- they are limited to a single domain, i.e., the design of WWW sites,

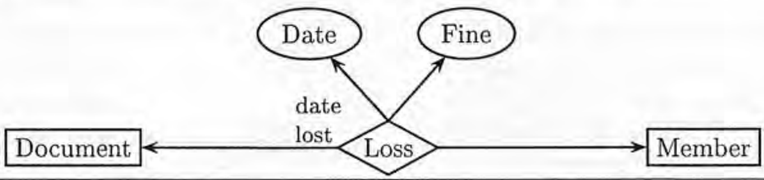
Figure 8.3 Entity Relationship model (adapted from Wieringa (1996, p. 176))



- they have been built on purpose to be used by our elicitation system.

One question that we can ask ourselves then is how would the system perform with an ER model that has been built independently? In order to evaluate this, we have adapted for use by our system an ER model described in Wieringa (1996). This model is presented in figure 8.3. It describes the entity-sets and relation-sets of importance for a university library. `Documents`, which are owned by `departments`, correspond to titles. `Titles` are described by their name and author. `Member` of the library can borrow documents (`Loan`). They can also reserve titles (`T-Res`), in which case they will be given any document corresponding to the title when one is available, and documents (`D-Res`), in which case they will be given a particular document when it becomes available. Finally, documents may be lost. In that case, members receive a `fine` (`Loss`). Our system cannot deal with this model as it is presented. This is due to the ternary `loss` relation between `document`, `member` and `fine`. Although this relation can be handled by the elicitation module, it cannot be dealt with properly by the dialogue manager. This is due to the simple naming process when

Figure 8.4 Transformed Entity Relationship model (partial view)



generating outputs (see section 5.4). During this process, the role of an entity in a relation is used to present this entity. For example, if we assume the role of `fine` is "charged", then the presentation of the loss relation from the `member` viewpoint would result in a sentence like "member #m1 is charged fine #f1". This is a satisfactory sentence. The problem is that the same role name is used when presenting the relation from the `document` viewpoint. This gives the sentence "document #d1 is charged fine #f1" which is incorrect. The same problem would appear for any non-binary relations. There is no simple way of dealing with this problem without changing the naming process used by the dialogue manager. This process would need to take into account the entities mentioned in the sentence in order to choose the correct name for the role. Rather than modifying the naming process we slightly modified the ER model to remove the ternary relation. We did this by making `fine` an attribute of the `loss` relation. We also added a minimum cardinality of 1 for documents. Although not equivalent to the initial model, it is close enough for our purposes. The new ER model is presented in appendix A.3. Using this model, we have been able to describe the state of a simple and imaginary library without difficulties. The conclusions we can draw from this experiment are:

- The natural language understanding and generation processes are too simple to deal with complex ER models,
- Other models than the one we created on purpose can however be adapted to fit in our framework without too much work.

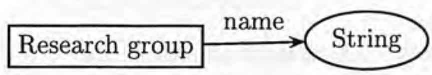
8.4 Domain model acquisition

A final aspect that we will investigate regarding the possibility of scaling up our system is the acquisition of a domain model. A possibility for acquisition has already been discussed in the previous section where we used an existing ER model as the specification model of our system. Another possibility is to write a model from scratch. This process is then equivalent to entering requirements. The difference is that the requirements do not describe constraints on a system, e.g., WWW site or library, but constraints on a specification model. The model on which the elicitation process is based on is then a specification "meta-"model which is simply the model underlying our ER representation (see section 4.2.1). This model contains elements such as `entity set` and `relation set` which, when instantiated, represent the domain model. It is presented in appendix A.4. An example of a model elicitation dialogue is given in example 8.2. An entity-set is created and is associated with an attribute. It corresponds to the model presented in figure 8.5 where the cardinality constraints have been omitted.

Example 8.2 Specification model elicitation dialogue

S1:	Entity set 1 is a entity set.
S2:	What is its name?
U1:	Research group.
S3:	What is its cardmin?
U2:	1.
S4:	What is its cardmax?
U3:	1.
S5:	Do you want an attribute attribute of it?
U4:	Yes.
S6:	An attribute is attribute of entity set research group.
S7:	Which attribute is attribute of entity set research group?
U5:	A new attribute.
S8:	Attribute 5 is attribute of entity set research group.
S9:	What is the name of attribute 5?
U6:	name.

Figure 8.5 Elicited model



As soon as a part of the specification model has been defined, it can be used to enter a specification. This is done by translating the result of the acquisition in the representation framework used for the specification model. This is a straightforward process since there is one-to-one mapping between the instantiated meta-model and the concepts in the domain model. This is shown in example 8.3 extending example 8.2. The dialogue moves between the specification and the specification model in exchanges U7 to S13.

Acquiring specification model from natural language is not new (Black 1987). Moreover, the limited natural language processing capability of our system makes the acquisition process sometimes quite stilted and less powerful than some other approaches where natural language cues are used to avoid questions. For example, the sentence "entity 1 *must* be related to entity 2" indicates that the minimal cardinality of the relation from entity 1 to entity2 is at least 1. By taking this piece of information into account, some questions about the relation cardinality may be avoided. However, the capability of our system to shift between the acquisition of the specification model and the statement of the specification itself in a focused way gives it more flexibility. Enabling the system to make use of linguistics cues to improve the elicitation process is mainly a matter of adding syntactic sugar to the interpretation templates.

The conclusions we can draw from this experiment are:

- Our system can be used for acquiring its own specification model,
- The acquisition process is not better than with other approaches but the capability to mix the different levels of acquisition (specification model and specification) in a

Example 8.3 Specification model elicitation and instantiation dialogue

S1:	Entity set 1 is a entity set.
S2:	What is its name?
U1:	Research group.
S3:	What is its cardmin?
U2:	1.
S4:	What is its cardmax?
U3:	1.
S5:	Do you want an attribute attribute of it?
U4:	Yes.
S6:	An attribute is attribute of entity set research group.
S7:	Which attribute is attribute of entity set research group?
U5:	A new attribute.
S8:	Attribute 5 is attribute of entity set research group.
S9:	What is the name of attribute 5?
U6:	name.
S10:	Do you want entity set research group having another attribute?
U7:	I want to speak about the research group.
S11:	OK. Research group 4 is a research group.
S12:	What is its name?
U8:	Software Systems and Processes
S13:	Do you want a role filled by entity set research group?

focused way gives some flexibility that other approaches do not have.

Chapter 9

Evaluation

Objectives

The objectives of this chapter are to:

- Evaluate the focus theories and conclude that local and global focus theories mixed together perform well,
- Present the use of automated users to carry out evaluations that would be too costly to carry out with human beings.

9.1 Introduction

In this section we present an evaluation of the impact of using focus theories. We evaluate the effect of the local focus theory and the global focus theory presented in chapter 3.

Numerous pieces of work make use of focus theories. The theories are sometimes modified to achieve specific aims such as marking some focus moves in the dialogue (e.g., Mittal et al. (1998); Huang (1994b)). However, most of these approaches do not precisely evaluate what the contributions of the focus theories are. The fact that they provide improvements in the dialogue quality is assumed. Moreover, focus theories, especially local focus theories, are often used for anaphora resolution. They are therefore evaluated on their power to disambiguate the references of pronouns. While this is a useful measure, it does not directly apply to our case where we need to evaluate the impact of focus theories on the perceived quality of dialogue organisation.

In our case, we wanted to see how the focus theories influence the capability of the system to produce coherent dialogue. The two questions we answer here are:

- Do focus theories provide improvements over other dialogue management strategies in perceived dialogue quality?
- How do focus theories differ in terms of improvements provided?

We will show that picking communications at random has a disastrous effect on perceived dialogue quality. Following the elicitation module order performs reasonably well. However, this result is quite unpredictable. Some focus theories, or combination of theories, improve the perceived dialogue quality noticeably compared to the elicitation module order. However, this is not the case of all focus theories. We also provide some explanations of the results found.

We now describe the setting in which the evaluation has been done. The evaluation is about the "attitude" aspect of the system, i.e., it is about the acceptability of the system by users by opposition to its effectiveness, learnability or flexibility (Lindgaard 1994, p. 30–31). As such, the evaluation requires users' involvement since acceptability is a subjective notion. We considered that users of the system should be accustomed to using computers but not our elicitation system. Therefore, we selected, for the experiments we carried out, subjects with some computer experience but without any particular knowledge regarding our system. Users were only given a minimal training on the system as it is supposed to be used easily. Users who had no linguistic knowledge were given short explanations on the evaluation they had to perform in order to ensure they understood exactly what was asked in the experiments. We took care of not giving any indications as to which dialogues we wanted them to prefer. As explained in chapter 4 (see figure 4.2), the focus theories intervene through the dialogue manager by helping select a communication from the communication pool. Communications are put in this pool by the elicitation module. The selected communication is then output in natural language. Since we wanted to test the theories in the same conditions, we chose to use the same elicitation module and the same output process for all tests. The elicitation system is a loose one, i.e., putting as many communications in the communication pool as possible (see section 4.5). Therefore, it offers many choices for selection and gives an important role to the dialogue manager (see section 7.4). The natural language generation process can run without focus information but changes the text output if information is available. The resulting experiments can be seen as a black-box experiment where the influence of the theories are evaluated through the behaviour of the whole system (Sparck Jones and Galliers 1996). All the experiments were run on a network of UNIX SUN workstations. The system is programmed in SICSTUS Prolog. It asks and answers questions in real time.

In all the following experiments, the present-first theory (see section 3.4.1) was used. It served as a minimal presentation strategy.

9.2 First evaluation

In the first evaluation, using a focus theory is compared to having no dialogue management strategy. It is indeed not useful to implement complex focus theories if they do not provide clear improvements over simpler natural language processing techniques. Therefore, the use of the theories should make a real change so that users can feel the difference when performing their task.

In order to evaluate whether focus theories have an influence on dialogue quality, we defined a global criterion of dialogue quality. This criterion is based on proposals made in the literature (Sikorski and Allen 1996; Smith et al. 1992; Smith 1997; Walker et al. 1997d,a, 1998). It is calculated by summing up answers in a questionnaire. Some questions and the aspects of the main dialogue quality they test are shown in table 9.1. The complete questionnaire is presented in appendix C.2. Questionnaires are powerful tools to obtain data if they have been carefully prepared. The one we use is similar to the those presented in the literature. Questions are not orthogonal, i.e., they may relate to the same aspect of the

Table 9.1 Questions and dialogue properties

Question	Quality aspect
Was it easy to state the requirements?	Natural language understanding quality
Was the dialogue related to the task you were performing?	Focus quality
Was the dialogue easy to understand?	Natural language generation quality
Was the order of the communications (presentations and questions) correct?	Coherence quality

Table 9.2 First evaluation (evaluations vary from 0 (very bad) to 16 (excellent))

	No management[a].	One focus theory	Two focus theories
Evaluations	2, 3, 5	15, 10, 6	10, 12, 12, 10, 8

[a]Communications are selected at random from the communication pool

dialogue. This is not a problem as we only considered the global quality criterion obtained from them. Each question (except the first and the last which are not evaluation questions) counts from 0 up to 4 in this final criterion, depending on the users' answer (the higher the user's satisfaction, the higher the mark).

The experiment then consisted of having users use our system. Users were asked to enter requirements about a given WWW site. The site consists of a home page linked to three pages. These pages display a navigator pointing to the home page. (A navigator is a group of hypertext-links.) The requirements, which have been expressed crudely on purpose, are presented in appendix C.1. The system was equipped with no focus theory, with either a local focus theory (see section 3.3) or a global focus theory (see section 3.2), or with two focus theories (both local and global focus theories). When no focus theory was used, communications were picked at *random* in the communication pool. Users had mostly to answer questions (the system had the initiative) but could also pass them, i.e., decline to answer them (see section 6.4.3). Other mixed-initiative possibilities were limited. Users could also ask to see the current state of the specification. Once they had finished the task (or when they thought they could do no better), users were asked to fill in the questionnaire about the dialogue they had had.

Eight users were asked to enter a given specification. Seven of the users had never used the system before. We obtained eleven dialogues. The experiment lasted less than two hours, including a presentation of the system to the new users.

The evaluations given by users are presented in table 9.2. Because some users produced more than one dialogue, we checked that the level of experience had no effect on perceived dialogue quality (correlation coefficient $r = -0.2229$). We then found that there was a statistically significant difference between the evaluation of a random dialogue, i.e., a dialogue where communications are picked by the system at random from the communication pool, and the evaluation of a dialogue using focus rules (Student test with the null hypothesis that there is no difference in evaluation means, $t = 4.15$ (the bigger t, the lower the probability of error in rejecting the null hypothesis); degree of freedom, $df = 9$ (the bigger df, the lower t needs to be to reach a given probability of error in rejecting the null hypothesis); probability of error in rejecting null-hypothesis, $p = 0.002$). This first experiment therefore confirms that focus theories can improve the perceived quality of dialogues.

We also tested whether the improvements in perceived dialogue quality were related to better specifications. Therefore, we searched whether the dialogue evaluation was related to the number of errors made while entering the requirements about the given site. Results point out that there may be a reduction of errors with better dialogues. However, there were not enough data to get statistically significant results (correlation coefficient, r = -0.1926; error on r, $\sigma_r = 0.2$; Student test with null-hypothesis 'r = 0', t = -0.95; degree of freedom, df = 9; probability of error in rejecting null-hypothesis, p = 0.3657).

9.3 Automated evaluation

Although the first experiment let us know that focus rules may be useful in dialogue production, it does not show that they perform better than other dialogue management strategies such as following the elicitation module order, i.e., selecting the latest communication produced by the elicitation module. Testing these possibilities would require a heavy investment in users' evaluation of the system. In order to reduce this investment while being able to test other dialogue management strategies, we designed an automated evaluation procedure.

9.3.1 Automation

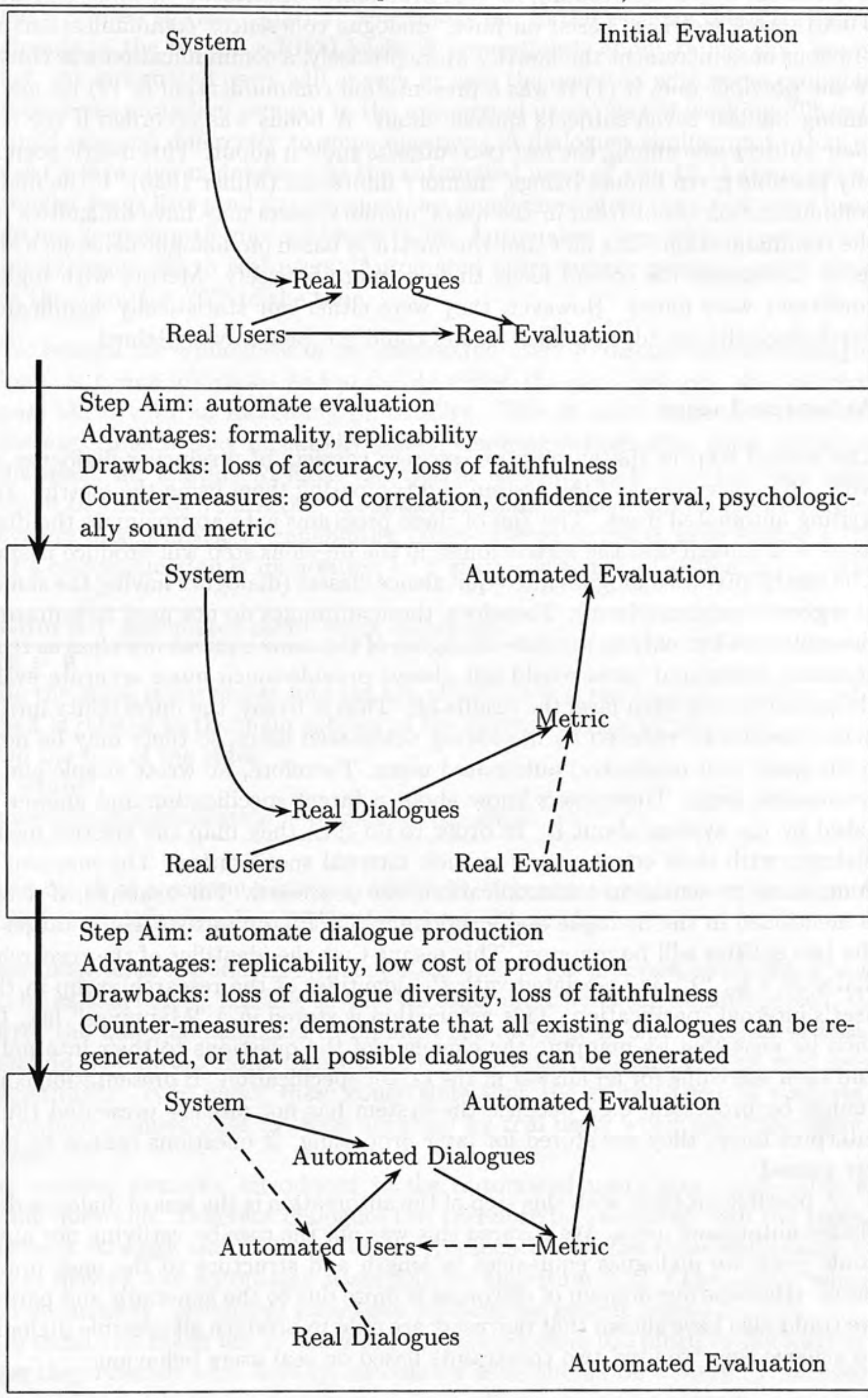
The aim of the automation is to replace the real users with automated users while obtaining an evaluation that remains faithful to the real users' perception of the dialogues. The automation of the evaluation is composed of two steps as shown in figure 9.1. They are described below. Each step consists of automating a part of the evaluation process. Each step provides gains but involves possible losses. Counter-measures are taken to limit these losses.

Metric

The first step consists of finding a metric based on easily measurable dialogue properties which is correlated with the evaluation given by users. The idea is that the metric should allow an easy and clear separation between the strategies which provide improvements in dialogue quality and those that do not. We assume that by using this metric we are able to evaluate automatically the quality of a given dialogue in the same domain. In other words, the metric replaces the questionnaire in the evaluation process. A possible problem posed by this step is the loss in accuracy and in faithfulness to the initial evaluation. To counterbalance this loss we search for a metric well-correlated with the evaluation and psychologically motivated. The good correlation ensures that the results we find are mathematically motivated. The psychological soundness serves as a filter to reject metrics based on unusual and hard-to-motivate measures. This helps us avoid over-tuning the metric to the evaluation sample we use.

Several metrics often proposed in the literature as being related to dialogue quality were recorded during the users' evaluation. We found that some obvious choices were not satisfactory in our case. For example, the length of the dialogue was not well correlated to the evaluation (r = 0.4019). The number of passed communications (r = -0.1207) and the number of times users asked to see the specification (r = -0.3595) were not good metrics either. We found that one of the metrics accounted for 65% of the evaluation mark (correlation

Figure 9.1 Automated evaluation (Solid lines show how the evaluation is carried out. Dashed lines show how automated components are obtained.)



coefficient, r = 0.8419; error on r, $\sigma_r = 0.2$; Student test with the null hypothesis that r = 0, t = 4.08; degree of freedom, df = 9; probability of error in rejecting null-hypothesis, p = 0.003). This metric is based on linear dialogue coherence: communications related to the previous ones increment the metric. More precisely, a communication was considered related to the previous ones if (1) it was a presentation communication or (2) its main subject was among the last seven subjects spoken about. A bonus was accorded if the communication main subject was among the last two subjects spoken about. This metric seems psychologically possible given human beings' memory limitation (Miller 1956). If the main subject of a communication is not fresh in the users' memory, users may have difficulties in interpreting the communication. The fact that this metric is based on dialogue coherence also encourages us to investigate the role of focus theories more precisely. Metrics with higher correlation coefficient were found. However, they were either not statistically significant or were not psychologically sound, i.e., their results could not be easily explained.

Automated users

The second step in the automation process consists of producing dialogues automatically with different versions of the system and comparing them using the metric. This is done by writing automated users. The aim of these programs is to approximate the dialogues of real users well enough that the metric found in the previous step will produce useful evaluations. The metric divides dialogues into equivalence classes (dialogues having the same metric value at a given confidence level). Therefore, these automata do not need to imitate all aspects of the real users but only to produce dialogues of the same equivalence class as real users. More accurate automated users would not always provide much more accurate evaluations since the metric would often level the results off. That is to say, the uncertainty introduced by the metric cannot be reduced by improving automated users, so there may be no need to have really good (but expensive) automated users. Therefore, we wrote simple and cost-effective automated users. These users know about a target specification and answer the questions asked by the system about it. In order to do this, they map the entities mentioned in the dialogue with their counterparts in their internal specification. The mapping is essentially done when presentation communications are processed. For example, if a research group is mentioned in the dialogue and a unique research group exists in the target specification, the two entities will be mapped. This means that the identifier of the research group in the dialogue, e.g., "1", is associated with the identifier of the research group in the automated user's internal specification. This association is stored in a "Mappings" list. Questions can then be answered by mapping the elements of the questions to their internal counterparts and then searching for an answer in the target specification. If presentation communications cannot be processed, e.g., because the system has not already presented things needed to interpret them, they are stored for later processing. If questions cannot be answered, they are passed.

A possible problem with this step of the automation is the loss of dialogue diversity due to simple automated users. We ensured this was not the case by verifying our automated users could generate dialogues equivalent in length and structure to the ones produced by real users. (Because our domain of discourse is finite due to the generator and parser limitations, we could also have shown that our users are able to produce all possible dialogues.) In order to achieve this, we used two constraints based on real users behaviour:

- We limited the memory span of our automated users so that they would forget things as our human subjects were observed to do. We approximate this by making the simulated users keep a list of things in focus. The list, called the "Focus" list in algorithm 9.2, has a limited storage capacity. New things put in the list may therefore displace those already in the list, on a FIFO basis. If a question is about things that are not in the list, the automated users will answer or pass the question with some probability. This introduces a random element in the automated users' way of working. Therefore, they could respond differently to some questions in dialogues similar up to that point. We used a focus list maintained by the automated users of size 10. Automated users with smaller focus lists tend to pass questions much more often than real users because they do not keep enough information in focus. Automated users with longer focus lists pass rarely compared to real users. Automated users answer questions with elements not in the focus list 70% of the time.

- We limited the willingness of our automated users to discuss the same subject for too long. If communications go too deeply about the same subject, the automated users pass them with an increasing probability. This is what some real users did during the experiment: they repeatedly passed communications after some subject had been discussed for a while to force the system to come back to a previous topic. (This could be done by redirecting the dialogue in the system but as mentioned earlier the system mixed-initiative capabilities during the experiment were limited.) The depth of a communication is its position on a stack computed as shown in algorithm 9.1. If

algorithm 9.1 Automated users' stack algorithm

Stack $\leftarrow \emptyset$
while the stack is not empty and the top of the stack is not a communication whose main subject is related to the main subject of the current communication **do**
  Pop the top of the stack
end while
The current communication is pushed on the stack

  the depth of a communication is above 10, it is passed with an increasing probability. This probability reaches 100% when the depth is 15.

The two constraints introduced have opposite effects: the first one forces the dialogue manager to be as coherent as possible so that its communications can be interpreted; the second one forces the dialogue managers to segment the dialogue in small chunks so that communications do not go too deep. This is consistent with the behaviour of real users in our first experiment. It is under these constraints that the metric used to evaluate the dialogue produces results close to those obtained by real users when using the same dialogue strategies.

The random elements introduced in the automated users give them some leeway in answering questions. Different dialogues can therefore be generated with the same dialogue management strategy and the same target specification by the same automated user. The algorithm driving the automated users follows algorithm 9.2. (The exact values for the random constraints (see above) have been replaced by generic tests.)

Care should be taken not to tailor the automated users too much to the kind of dialogue manager they interact with. Ideally, automated users should be written by someone with no knowledge of how the dialogue manager works. This was not possible for practical reasons

algorithm 9.2 Automated users' algorithm

```
Communication stack ← ∅ {contains the communications related to the current topic and
acts as a stack}
Focus list ← ∅ {contains the things spoken about recently and acts as a FIFO}
Mappings ← ∅ {contains the mapping between dialogue entities and their internal coun-
terparts}
Unprocessed Communications ← ∅ {contains the presentation communications that could
not be processed immediately}
while there is a communication to process coming from the dialogue manager do
  while the communication stack is not empty and the top of the stack is not a commu-
  nication whose main subject is related to the main subject of the current communication
  do
    Pop the top of the stack
  end while
  Push the communication on the stack
  if this is a presentation communication then
    if it can be interpreted using Mappings then
      add any new mappings to Mappings
      add its subjects to the focus list
      repeat
        select a communication from Unprocessed Communications
        if it can be interpreted using Mappings then
          add any new mappings to Mappings
          remove the communication from Unprocessed Communications
        end if
      until no communication can be removed
    else
      add the communication to Unprocessed Communications
    end if
  else
    if a random number is greater than the communication depth then {this is where a
    random element is introduced}
      if it can be interpreted using Mappings then
        if its subjects are in the focus list then
          answer the question
          add any new mappings to Mappings
          add its subjects to the focus list
        else
          if a random number is greater than a given threshold then {this is where a
          random element is introduced}
            answer the question
            add any new mappings to Mappings
            add its subjects to the focus list
          else
            pass the communication
          end if
        end if
      else
        pass the communication
      end if
    end if
    pass the communication
  end if
end while
```

in this thesis, but the automated users development process was kept clearly separated from the dialogue manager development process.

Evaluation

Finally, we can run the system with the automated users and obtain an evaluation of new dialogue management strategies. (A sample of a dialogue generated automatically is given in appendix B.2.) The results given by this approach cannot be very accurate since the use of a metric and automated users introduces approximations. However, these losses in accuracy can be partially evaluated and counterbalanced. For example, by considering confidence intervals rather than point evaluations, we can get a better opinion on the real differences between strategies. Confidence intervals can be computed from the uncertainty on the automated users' evaluation due to the random element (This is evaluated by running automated users several times in the same setting and computing the variation in the evaluation.) and the uncertainty on the metric due to the imperfect correlation with a real evaluation. (This is computed from the correlation coefficient obtained when the metric is found.)

Two main assumptions are made during the automation process:

1. A well-correlated and psychologically sound metric can be found. Numerous metrics have been studied and evaluated in the literature. However, this may still be difficult for dialogue aspects on which little research has been carried out.

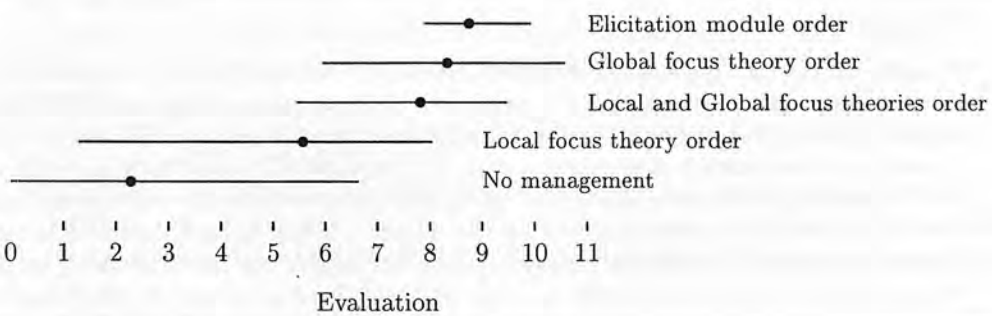
2. Automated users representative of real users can be written. This is usually possible since the system under study can only accept a limited number of answers. Therefore, users (real and automated) are restricted in what they can express. However, this may still be difficult to achieve if, for example, users need a lot of common-sense knowledge to interact with the system.

Not all new dialogue management strategies can be tested in this way. The use of a metric puts constraints on the type of dialogues that can be evaluated. For example, the length of the dialogue was shown not to be a good metric in our case. However, this metric could be good for other dialogues, e.g., if the dialogue manager's main role was to prune communications rather than to order them. If the type of dialogues evaluated changes, a new metric should be used. The same pertains to automated users. These users have been written to have a dialogue with a particular type of dialogue manager. If the latter changes too much, automated users may not be able to deal adequately with it. For example, if automated users are not programmed to make use of cue words (such as ours), they won't distinguish between a dialogue manager outputting useful cue words and a dialogue manager not outputting any cue words. It is therefore important to compare dialogue managers that are not "radically" different.

9.3.2 Results

Using the metric and the automated users, we were able to test if other dialogue management strategies, that were not tested during the first experiment, could also provide improvements in dialogue quality. One such strategy was to simply follow the elicitation module order (in effect by-passing the dialogue manager and considering the communication pool as a FIFO store). This strategy turns out to perform slightly better than focus theories. This surprising result comes from the way our elicitation module works. By always outputting

Figure 9.2 Evaluations (vary from 0 (very bad) to 16 (excellent)) and 90% confidence intervals



all the questions it can on the current state of the specification, it follows an implicit task model based on the domain model. We have seen, in sections 2.3.1 and 2.3.2, that dialogues based on these models are usually globally coherent. This may explain why these dialogues seem good. However, this would not be true for any elicitation system in general, which by-default do not manage dialogues. Some may produce bad dialogues (possibly as bad as the random order). Note that having an explicit model of the task would help understand how the system works but this would not make up for the lack of dialogue management knowledge.

We also tested the theories presented in chapter 3 separately. The evaluations are shown in figure 9.2. The local focus theory, although better than the no management policy where communications are picked at random in the communication pool, performs less well than the global or elicitation order strategies. This could be due to the fact that a local dialogue management leads to "spaghetti" dialogues (Sibun 1992) which are not optimal in our case.

Based on these results, four strategies were selected for further evaluation: elicitation module order, global focus theory, local focus theory and the mix of global and local theories. The strategies were selected because they form the best cluster in the evaluation and we have found reasons to reject the other strategy. We could have eliminated the local focus theory as well based on the fact that it seems to perform poorly with automated users. We keep it however in order to study further its effect on perceived dialogue quality. The issue of eliminating theories when developing a dialogue manager is dealt with again in chapter 10.

9.4 Final evaluations

In the following evaluations, we evaluated the theories selected in the previous section by asking people to evaluate dialogues produced with or without them. We did not search for a correlated metric (although we recorded metric values in case of further development) and we did not develop automated users.

9.4.1 Global coherence

We first tested if the strategies were ensuring good global coherence. We tested this hypothesis using the Analytic Hierarchy Process (Saaty 1990). This process can be used to rank alternative situations, in our case dialogues, by making pair-wise comparisons between them. People read transcripts of dialogues generated with and without dialogue strategies and then compare the dialogues pair-wise based on their global coherence. A dialogue was defined as globally coherent if it is divided into well defined chunks that are correctly related together. Coherence was presented on a few dialogue examples. Pair-wise comparisons are carried out by filling a square matrix with the comparison mark of dialogues i and j in cell (i, j). (In fact only $n(n-1)/2$ comparisons need to be made. The rest of the matrix can then be filled in automatically.) The marks vary from 1 to 9 if dialogue i is better than dialogue j (the higher the mark, the better dialogue i is compared to dialogue j) and from 1 to 1/9 if dialogue j is better than dialogue i. (These numbers are given in Saaty (1990)). Transformations can then be made on this matrix to compute the overall ranking of the strategies (see below). The experiment sheet is presented in appendix C.3.

Table 9.3 Comparison grid for global coherence

	Local and Global theories	Global theory	Elicitation module	Local theory
Local and Global theories	1	1/2	4	4
Global theory	2	1	4	4
Elicitation module	1/4	1/4	1	2
Local theory	1/4	1/4	1/2	1

Table 9.4 Theory ranking by global coherence

Local and global theories order	33
Global theory order	47
Elicitation module order	12
Local theory order	8
	100

The experiment involved six persons and lasted 90 minutes. Six dialogues were evaluated. Two dialogues were based on the elicitation module order, two on the global focus theory, one on the local theory and one on local and global focus theories combined. All dialogues were about the same specification. Each participant compared four dialogues, resulting in 36 pair-wise comparisons. The results, after averaging the users' answers, are given in table 9.3.

Pair-wise comparisons can be transformed in an overall ranking of the alternative strategies (Saaty 1990). This is done by computing and normalising the principal eigen vector of the comparison matrix. Its coordinates give the relative importance of the different strategies. For example, a theory with a value of 60 is considered three times more globally coherent than a theory with a value of 20. The values of all the theories add up to 100. The results are presented in table 9.4. (Consistency ratio = 0.0818. A consistency ratio less than 0.1 indicates reliable results.) They clearly show that following the elicitation system order performs poorly compared to using a global focus theory (3 to 4 times worse than a focus theory). This result, which is surprising given the results of the first evaluation, may be explained by the fact that the sentence microplanning, e.g., choice between active and passive and pronoun generation (see section 5.4), cannot be performed without focus information. The metric and the automated users used in the previous evaluation do not take this aspect into account and rank the elicitation module order highly purely based on the text flow. However, when compared pair-wised by human subjects with dialogues where focus theories allowed this microplanning to be carried out, the elicitation module dialogues seem poorer. This suggests a way of improving latter evaluations by taking surface phenomena into account. The results also confirm that local theory is performing badly as was predicted in the previous stage. The local and global theories mixed together are performing less well than the global theory alone. This can be explained by the local theory smoothing the transitions between focus spaces. Therefore, the dialogue seems less well divided than with the global theory only, although the focus spaces are equivalent.

9.4.2 Local coherence

Finally, we tested the theories' capability to improve local coherence. A dialogue was defined as locally coherent if it has well related successive sentences. We tested this hypothesis with the same kind of experiment as in the previous evaluation.

The results are presented in tables 9.5 and 9.6 (Consistency ratio = 0.0993). The theory mix consisting of a local and a global theories is performing better than the global theory alone. However, the difference is not big. This can be explained by the small size of the focus spaces. Since just a few communications are output in each space (around 4 on average), a suboptimal coherence strategy does not decrease the perceived local coherence too much. A more suprising result is the poor performance of the local theory (and to some extent the performance of the elicitation module which basically follows a local strategy). This may be explained again by the spaghetti-like nature of the dialogues obtained with these dialogue management strategies. Although locally coherent, the dialogue becomes hard to read and is therefore badly rated. More experiments are needed however to determine if this is indeed the source of the bad mark.

Table 9.5 Comparison grid for local coherence

	Local and Global theories	Global theory	Elicitation module	Local theory
Local and Global theories	1	1	5	2
Global theory	1	1	3	3
Elicitation module	1/5	1/3	1	4
Local theory	1/2	1/3	1/4	1

9.5 Conclusion

From this analysis, we can conclude that, in our domain, the local and global theory mix and the global theory on its own provide improvements in dialogue quality and especially

Table 9.6 Theory ranking by local coherence

Local and global theories order	39	━━━━━━━━━━━━━━━━━━━━
Global theory order	34	━━━━━━━━━━━━━━━━━
Elicitation module order	17	━━━━━━━━
Local theory order	10	━━━━━
	100	

in dialogue coherence. The global theory performs better on global coherence as can be expected and the theory mix performs well on local aspects. This good performace can be explained by the complementarity between the local and global theories (see section 7.4). It should be noted that the local theory is not performing well on its own. This theory seems to perform well in limited spaces (e.g., when a focus space has been defined by a global focus theory) but less and less well as the spaces become larger. This shows that the use of focus theories does not always provide improvements in perceived dialogue quality compared to simpler stategies. A precise understanding of the properties of a theory and an analysis of the conditions under which it operates are needed to estimate its influence on dialogue quality. Whether or not this can be formalised and therefore serve as a criterion for choosing the "right" theory for a given application remains an open issue (see section 11.2.5). The elicitation order which was not at first considered has also been introduced at a very low cost in the evaluation process, but was later found to have a poor performance in global and local coherence compared to some focus theories. The fact that this order is not as formalised as the focus theories are makes it difficult to predict the properties of the dialogues produced by following it.

This analysis was performed on a limited domain. However, the evaluation was not dependent on the domain. We therefore believe that the results have a wider application than the requirements engineering domain.

[illegible] some dialogue quality components [illegible] worse than [illegible] the results of the first evaluation, may be explained by [illegible] the sentence [illegible] choice between active and passive [illegible] (see section [illegible]) [illegible] performed without focus information. [illegible] and the parameters used [illegible] evaluation do not take this aspect into account and [illegible] the elicitation module order highly purely based on [illegible] dialogues where [illegible] the elicitation module dialogues [illegible] surface phe- [illegible] badly as was [illegible] together are performing [illegible] theory smoothing [illegible] divided than [illegible] defined [illegible] hypothesis with [illegible] theory [illegible] global theory [illegible] the focus space. [illegible] average), [illegible] local coherence too much. [illegible] the performance of the elicitation module [illegible] This may be explained again by the [illegible] nature of the dialogues obtained with these dialogue management strategies. Although locally coherent, the dialogue becomes hard to read and is therefore badly rated. More experiments are needed however to determine if this is indeed the source of the bad marks.

Table 9.5: Comparison grid for local coherence

	Local and Global theories	Global theory	Elicitation module	Local theory
Local and Global theories	-	1	2	2
Global theory	1	-	3	3
Elicitation module	2/3	1/3	-	4
Local theory	1/2	1/3	1/4	-

9.6 Conclusion

From this analysis, we can conclude that, in our domain, the local and global theory mix and the global theory on its own provide improvements in dialogue quality and especially

Chapter 10

Development Method

Objectives

The objectives of this chapter are to:

- Present the need for regular evaluation when developing dialogue managers,
- Present the use of automated users to perform evaluation at a minimal cost,
- Describe a possible development cycle and an example of its application to our system development.

10.1 Introduction

In this chapter, we present a development method we have used to build our system and to produce the results presented in chapter 9. This method is based on the use of automated users to regularly evaluate the system under development and guide its construction. The method presented here could be the basis for a new and generally useful development approach for interactive research systems.

Because there is not yet a clear method for the development of dialogue managers, an iterative approach using evolutionary prototypes seems adequate (Partridge 1992; Sommerville 1995). Each iteration is a step towards the final program. However, not going astray is a major difficulty. Evaluating whether a dialogue management strategy is beneficial or not, in terms of performance but also in terms of cost-effectiveness, is therefore important.

We present some existing evaluation methods that can be used to evaluate a dialogue manager in section 10.2. Most of these approaches require a lot of work which makes them unsuitable for repeated evaluations during system development. We then describe how to integrate the automated users presented in chapter 9 in a development method allowing frequent and cost-effective evaluations when building a system.

10.2 Evaluation methods

Evaluation of dialogue management strategies is a difficult task. Several sources can influence the dialogue beside the management techniques, and dialogue can be evaluated along several dimensions. A lot of current approaches to evaluation require heavy involvement from users (Sikorski and Allen 1996; Walker et al. 1997a). This makes them difficult to use during development. Some evaluation methods are based on automated users:

- Walker (1996a) uses automated users whose dialogue strategies and memory and reasoning capabilities can be adjusted. Communication and reasoning have an associated cost. Dialogue performance is then calculated by considering the overall cost of the dialogue and how successful it was. The success of the dialogue is calculated from the degree of achievement of the task automated users had to carry out. This framework is useful for studying the theoretical implications of dialogue strategies depending on cognitive constraints (Hanks et al. 1993). However, it is unclear what the relation is between the dialogue performance computed and a real dialogue evaluation. (For example, counting the numbers of utterances may be less representative of the dialogue quality than of the elicitation system's capability to guess some of the specification.) Therefore, the application of this framework to dialogue quality evaluation is limited.

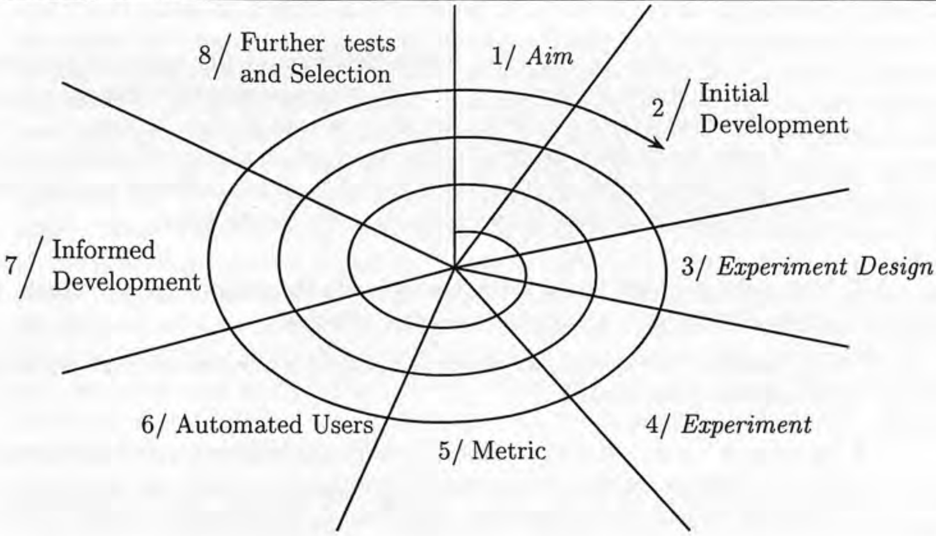
- Eckert et al. (1997, 1998) also uses automated users whose dialogue strategies can be adjusted. The adjustments are made by studying a corpus of existing dialogues. Dialogue quality is then calculating by using a metric which is chosen to be well correlated with real users' evaluation (Walker et al. 1997d). This framework is useful to test different dialogue strategies on a particular aspect of the dialogues and for specific users. However, since only one metric is used, it is not clear how to test the dialogues on different aspects, such as global and local coherence. It is also unclear if investing a lot of effort in accurate automated users is worthwhile when the dialogue manager is undergoing modifications which is the case during development. New automated users may be needed as the dialogue manager evolves (see section 9.3.2). It may therefore be better to write simpler automated users adapted to the metric used to test a particular aspect of the dialogues. The application of this framework during the development phase where the dialogue manager evolves and several aspects of the dialogues need to be evaluated seems therefore limited.

- Ishizaki (1997) also makes use of automated users to evaluate the conditions under which mixed-initiative dialogues are more effective than non mixed-initiative dialogues. In this case, a simple metric, consisting of the number of characters in the dialogues, is used. The automated users are also simple and only roughly based on existing dialogues. However, a mathematical model gives a sound theoretical basis to the study and the dialogue simulation with automated users is only used to confirm the theoretical findings. Unfortunately, such an approach is difficult to apply in our case since the notion of coherence is difficult to formalise and to study mathematically.

10.3 Development and test cycle

The main objective in our approach is to provide quick but faithful evaluations on particular aspects of the system. This is done by a light-weight evaluation process. Compared to the last two approaches presented above, we make fewer assumptions about the possibility of studying the dialogues statistically or analytically. This enables us to deal with more types of dialogues. On the other hand, we are not able to benefit from proven mathematical results which means that we have to ensure the correctness of our evaluation by repeating

Figure 10.1 Development and Evaluation Cycle



experiments. The process can be used while developing the system and can provide guidance on which aspect of the system should be worked on.

The development and test process is based on a spiral model of software development (Boehm 1987). The cycle we propose is composed of a cycle shown in figure 10.1 and described below. Each turn of the cycle represents the development and evaluation of one aspect of the system. The radial movement therefore represents the amount of testing carried out on the system. For each step we relate the task described to what we did during our system development and testing (see chapter 9).

1. The first step consists of defining the aim of the current cycle turn. This step has one of two results:

    - The cycle is stopped. The system has reached its objective or there is no cost-effective way of making progress.

    - A possibility of improvement is identified. Knowing which improvements should be attempted and tested first depends on the final requirements for the system, on the perceived risk in achieving some progress and on the results of previous cycles (see section 9.4.1 for an example where previous analyses suggest new aims). More important and more risky aspects of the system should be considered first since they are likely to influence heavily which dialogue management strategies can be used. The constraints under which the improvement should be obtained and a criterion are then defined. The improvement to be achieved is expressed in terms of that criterion. The criterion should therefore be testable since we will use it to evaluate the progress made. This ensures that the improvement can be measured and verified (Sommerville and Sawyer 1997, good practice 8.7). The aim of this criterion is to allow the partition of the possible additions to the system into those that improve the performance of the system and those that do not.

Application: In our case, in the first turn of the cycle we considered whether dialogue management was providing improvements in dialogue quality. We did therefore define a criterion for dialogue quality (see section 9.2). In the second turn, we studied the effect of management on global coherence (see section 9.4.1) and in the third we studied the the effect of management on local coherence (see section 9.4.2). Finally, we stopped in the fourth turn.

2. Next, development is carried out to provide the improvement wanted. Several competing strategies can be used to achieve this goal.

   Application: In our case, we developed several strategies, such as those based on local and global focus theories.

3. During this step, we design the evaluation that will be carried out in this turn of the cycle. Users' evaluation is necessary since there is usually no other way of knowing if a system is performing correctly: " in general, the straightforward notion of decidably correct or incorrect system behaviour even for specific inputs is difficult to pin down for many AI problems" (Partridge 1992, p30–31). We found that two kinds of experiments are especially useful for evaluation:

   - Interactive experiments where users play with the system and then answer a questionnaire. In general, questionnaires are a cost effective way of obtaining subjective data provided that they are carefully designed and have a clear purpose. This is the case here since we ask questions only on the aspect defined in step 1 and therefore reduce the risk of producing data that cannot be analysed.

   - Static experiments where users do not directly interact with the system but judge some of its outputs. Pair-wise comparison techniques (Karlsson 1996; Saaty 1990) can be used to compare different strategies, usually reducing the number of experiments to be done.

     Application: In our case, we chose an interactive experiment in the first turn of the cycle. We then chose static ones for the second and third turns.

4. The next step consists of the actual evaluation of the system by users. This is done by asking users to complete the experiment designed in step 3. At the same time, dialogues are recorded and metrics are measured. This step provides two results:

   - It enables us to know if there is a need for further evaluation. If no improvements can be gained then there is no need to go further.

   - Assuming that there are improvements, we can automate the evaluation. This is the role of steps 5 and 6.

   Application: In our case, we found that some management strategies had an effect on the aspect we were studying.

5. A major problem in obtaining a cost-effective development method is the need to automate the evaluation step because evaluation by users is time-consuming. As explained in chapter 9, if we can find a metric correlated with the results found in step 4, then we can use this metric in place of the questionnaire given to the users. The idea is that the metric should allow an easy and clear separation between the strategies which provide improvements and those that do not. Finding this metric is a difficult part of the process. Some obvious metrics such as number of turns in the dialogue or elapsed time may not always correlate well with the perceived dialogue quality (Walker et al. 1998). Metrics should be kept as simple as possible especially during the first cycles of the approach. Metrics should be based on dialogue properties and not on properties of the particular strategies used to generate the dialogues since the comparison of the internal working of different theories could be difficult or even impossible (Sparck Jones and Galliers 1996). Note that the metrics need not explain the results but just be correlated with them. The difference between metrics and the criterion defined in step 1 is that metrics are measurable, i.e., are objective, whereas the criterion is testable, i.e., depends on users' evaluation.

   Application: In our case, we found a metric during the first turn of the cycle that accounted for 65% of the users' evaluation. We did not search for a metric during the next turns as we were not developing new strategies.

6. The optional step consists of writing automated users. This step should be performed when users were directly experimenting with the system. Otherwise, it can be ignored. The aim of these programs is to approximate the dialogues of real users for the metric of step 5. Knowing which characteristics of the dialogue should be replicated by the automated users makes it easier to program them without having big corpora for evaluation. The good matching of the automated users with the metric should also be verified since some constraints may exist for the automated users/metric pair to provide good evaluations (see section 9.3.2). Once such automated users have been written, we can use them in place of real users. This is a big advantage as real users do often not have time to test a system under development. In particular, this enables us to test strategies that were not evaluated by real users (for example because they were developed later). This is the aim of the next step.

   Application: In our case, we developed, during the first turn of the cycle, simple automated users answering or passing the system's communications.

7. In the next step, new strategies are developed by taking into account the insights provided by the experiment. This informed phase of development is better focused on the problem and should result in better strategies.

   Application: In our case, this step was performed during the first turn of the cycle. It mainly consisted of debugging the dialogue management strategies. Because of the availability of automated users, debugging could be made incremental and was therefore easier than having to correct the system without feedback. (This was also seen as a main advantage of using automated users in Eckert et al. (1998).)

8. In this last step, we use the automated users of step 6 to produce dialogues and we evaluate them using the metric of step 5. We can therefore test different dialogue management strategies developed in the previous step at a minimum cost. If we want

to test a new strategy, we can do it without requiring real users evaluation. Care should be taken however not to tailor the strategies to the particular tests carried out. We have to ensure that the new developments are well-tuned but not over-specified for the tests used. It is also possible to test the new strategies with automated users created during previous cycles. This allows us to check that the latest modifications do not make the system regress in areas already tested.

Once strategies have been selected for further development, they are fed back to step 1 and the cycle recurs.

Application: In chapter 9 we kept all the main strategies for further evaluation since our aim was to study their influence on various aspect of perceived dialogue quality. Moreover, even when testing all the theories, the experiments needed to evaluate the system took only 25 person-hours on aggregate. However, in a normal development process where development is more expensive, we could have eliminated several dialogue management strategies along the way. In particular, we could have eliminated the local focus theory immediately after the first turn because it performed poorly for overall perceived dialogue quality (see table 9.2). The elicitation module order could have been eliminated after turn 2 because it performed poorly on global coherence which was judged an important factor for success (see table 9.4). We would have been left with the theories that turned out to be best. It should be noted that each new cycle is not necessarily concerned with a subset of the previous turns' concerns. An orthogonal issue can be dealt with. However, if we dismiss dialogue management strategies as soon as possible, more important issues should be dealt with first. Then only the most important strategies will be ranked during the next turns. If a minor issue is dealt with first, an important strategy could wrongly be eliminated. This is why we tested our strategies on issues of decreasing importance from first to third experiment.

Not all steps in the cycle are performed at each turn. A full turn is carried out during development. However, during evaluation only, i.e., when no new strategies are created, the cycle is reduced to steps 1, 3 and 4 (typeset in italic in figure 10.1). This is particularly the case towards the end of development when the emphasis change from development to testing. It is a good idea to terminate the process by some pure evaluation turns (as was done in our work as shown in table 10.1).

The evaluation approach presented here can be used during development. The low cost of this method enables us to screen many different dialogue management strategies to pick out the best. These strategies can then be subjected to more in-depth evaluation. However it can also be used to test full-blown systems. In this case, each cycle test a particularly important point of the systems so that their strength and weaknesses are evaluated. The evaluation stops as soon as one of the systems can be selected.

Table 10.1 Applying the development and test cycle for our system

		Cycles			
		1	2	3	4
Steps	1	Issue: quality improvements	Issue: global coherence	Issue: local coherence	Stop
	2	Dialogue managers development (e.g., with local and global focus rules)			
	3	Interactive experiment design	Static experiment design	Static experiment design	
	5	Metric			
	6	Automated users development			
	7	Further development and debugging			
	8	Further evaluations and selection	Selection	Selection	

Chapter 11

Conclusion and Further Issues

Objectives

The objectives of this chapter are to:

- Present what has been achieved in this thesis,
- Present issues left open for further research.

11.1 Conclusion

We conclude this thesis by discussing the main questions we asked in the introduction (see chapter 1).

- Which focus theories should we use?

  After reviewing the main focus theories in chapter 2, we have proposed two formalisations of focus theories in chapter 3. These theories are representative of two different classes of theories: one is a global focus theory which is concerned with the high-level organisation of the dialogue; the other is a local focus theory which is concerned with the sentence-to-sentence coherence of the text. Special attention has been paid to make these theories suited for natural language generation. We also made them as formal as possible so that some of their properties could be proved and so that they could be implemented. Other theories could be implemented but some, e.g., those based on semantic relations, are more difficult to formalise than others.

- What are the relations between the constraints imposed by the focus theories and the constraints inherent to the requirements elicitation process?

  We have found two kinds of dependencies between the constraints imposed by the focus theories and the constraints inherent to the requirements elicitation process:

  Representational dependencies which are due to the differences of representation frameworks used by the focus theories and the elicitation module. In our system, we tried to limit the amount of engineering work needed to apply the focus theories. Therefore, we re-used for the dialogue manager the domain knowledge used by the elicitation module. The representational dependencies are then formalised in translation rules which bridge the gap between the two representations. This is explained in chapters 3 and 4. As noted in these chapters, writing translation rules is a delicate problem which may have a big influence on the performance of the focus theories.

Functional dependencies which are due to the necessary interaction between the focus theories (through the dialogue manager) and the elicitation module. The way these two factors influence the requirements elicitation dialogue is presented in chapter 5 for the generation of outputs, in chapter 6 for the interpretation of inputs and in chapter 7 where a complete example makes this dependency salient.

- Does this approach improve the perceived quality of the dialogue between the elicitation tool and its users?

  As presented in chapter 9, the use of focus theories in structuring requirements elicitation dialogues may provide an improvement in perceived dialogue quality. However, not every focus theory acts in the same way. We have shown that global and local focus theories differ in the manner they influence the dialogue. We have also shown that using focus theories does not always provide improvements in dialogue quality compared to other dialogue management strategies. However, when the right focus theories are selected, they allow the system to produce better dialogues. Selecting the right focus theories is left as an open issue (see section 11.2.5). However, a mix of global and local focus theories seems to be a reasonable choice in most situations.

  It also remains to be seen whether these improvements in perceived dialogue quality translate into better specifications.

Two other points have been discussed in relation to our system:

- The capability of our system to scale up and its capability to apply to some other domains (see chapter 8). Despite limitations, which are mainly due to the very simple way our system handles natural language, focus theories seem applicable to different and/or more complex domains that the one we presented in our examples.
- A development method for dialogue managers based on automated users (see chapter 10). This method enabled us to carry out numerous evaluations at a very low cost.

11.2 Further issues

In this section we discuss some issues that we mentioned in the thesis but are still much open to further research.

11.2.1 Adaptive focus theories

Most focus theories assume a single strategy when applying the focus rules. For example, the centering theory always prefers retaining moves to smooth-shift moves while our local theory formalisation makes the reverse preference. However, it is unlikely that a particular move can always be preferred over another in all situations. If the dialogue has just started it may be preferable to give priority to smooth-shift moves as they introduce new things in the dialogue. On the other hand, if the dialogue is getting too complex, preferring retaining moves, which keep the same thing in focus, may be better. This is also true at a more global level. Always preferring additive moves over pop moves may not lead to an optimal dialogue, especially if the dialogue is going too "deep" and participants are forgetting the dialogue's main aim.

The problem is not so bad, as our system allows users to redirect the dialogue. If users judge that the topic chosen by the system does not correspond to what they want to speak about, they can always request a change. However, it would be interested to minimise the need for such requests. A major problem then is to know when to change the preferences for the different focus moves. Possible ways of measuring a dialogue evolution are for example the size of focus spaces, the number of controlling spaces, the number of communications with the same focus, the number of forward-looking centres introduced. However, even assuming that one of these measures reflects correctly the state of the dialogue, it is not clear what criteria should be used to change the dialogue management strategy.

11.2.2 Expectations

It is quite usual in dialogues to say up-front what will be spoken about. This was for example the case in some human dialogues about WWW site design we recorded. One of the participants would describe the main elements of the site such as publications, researchers and projects, and the dialogue would then deal with each of these topics in turn. It seems that this feature could be beneficial for dialogue managers as it leads to more structured dialogues. Explicitly mentioning how the dialogue will evolve enables participants to know what is important or not and to focus on things accordingly. This could serve as a way of organising the cache model to retain things we know will be spoken about (see section 2.3.4). One way of achieving this kind of behaviour is by presenting all the additive moves that can be done at some point in the dialogue, e.g., "we will speak about...", and then discussing each point separately, ensuring that we do not cross over into another topic. There are however numerous problems left such as: Where do we draw the boundary between topics? What do we do if we have to cross a boundary anyway? What are the linguistic cues used to introduced a topic? More research is needed to understand these phenomena.

11.2.3 Information exchanged

An issue related to the previous one is to know how much and what sort of information should be exchanged between the elicitation module and the dialogue manager. In our system, the communications contain a minimum of information. This makes the role of the elicitation module, which is concerned with reasoning on the specification, and the role of the dialogue manager, which is concerned with organising the dialogue, clearly distinct. This has some drawbacks though. As seen in the previous section, it could be beneficial for the reasoning module to inform the dialogue manager of the main subjects that will be dealt with. At a more local level, the dialogue manager would benefit from knowing which communications (or at least having some partial pieces of information on the communications) will be created next when a communication is processed. This would help process complex users' inputs. For the moment, the dialogue manager has a built-in knowledge of how to process these communications, which assumes a specific processing by the elicitation module (see section 6.4.2). There is therefore a lack of generality on our approach. Many communication protocols between agents, such as KQML (Labrou 1996; Labrou and Finin 1997), suffer from the same problem: only speech acts are exchanged, one at a time, without information on the overall dialogue structure. Some communications may restrict which communications are allowed next, e.g., a question (`ask-if` speech act in KQML) is expected to be followed by an answer (`tell` speech act in KQML) or an error message (`error` or `sorry` speech acts in KQML). However, these restrictions are not always sufficient to define precise expectations on the coming dialogue. More work needs to be carried out in order to know how these protocols could be extended to enable better dialogues.

11.2.4 Multiple message processing

Our system currently outputs messages one by one. There are two reasons for this behaviour: (1) it allows the elicitation module to change the content of the communication pool at any point in the dialogue and (2) it enables users to redirect the dialogue whenever they want to. This approach has some drawbacks however. For example, successive messages are generated independently, which may lead to unnatural texts. In some cases, several messages could certainly be output together. For example, series of presentation messages about a particular aspect of the specification could be output as a group. This would enable the dialogue manager to use more advanced natural language techniques, such as aggregation, to improve the text produced. This would not affect the freedom given to the elicitation system to change the content of the communication pool and users would still be able to redirect the dialogue. Enabling the dialogue manager to select more than one communication requires to answer some open issues such as how many communications to select. The problem is not to select one communication anymore but the most relevant and coherent subset from the communication pool.

11.2.5 Selecting focus theories

A final issue which requires more research is the selection of focus theories adapted to given types of dialogues and domains. As we have seen in chapter 9, the local focus theory we propose is not good for organising big focus spaces. It needs a relatively small space to operate efficiently. Finding ways of formalising constraints on the optimal use of focus theories is an important issue. It is however unclear what kind of criteria should be used to classify the dialogues types and domain types in which the theories will have to apply. Dialogues types may for example involve tutorials, where it could be better to deal with a topic thoroughly before moving on, or descriptions, where we may want to change the topic regularly to avoid lengthy paragraphs. Domain types could be classified for example by their connectivity, i.e., the ease with which we can go from one topic to another. If the domain is loosely connected we may prefer to push new topics as soon as we can, since we are not sure to be able to do it again. On the other hand, we may postpone speaking about a new topic if we are almost assured we can come back to it easily. These differences would lead to the selection of focus theories, or versions of the same theory, which are best adapted to the dialogue situation.

Bibliography

James Allen. *Natural Language Understanding.* Benjamin/Cummings Publishing Company, Menlo Park, California, USA, 1987.

James Allen and Mark Core. *Draft of DAMSL: Dialog Act Markup in Several Layers*, 1997.

J. L. Austin. *How to do things with words.* Oxford University Press, 1962.

Robert Balzer, Neil Goldman, and David Wile. Informality in program specifications. *IEEE Transactions on Software Engineering*, SE-4(2):94–103, 1978.

John A. Bateman. Upper modeling: A general organization of knowledge for natural language processing. In *Proceedings of the Workshop on Standards for Knowledge Representation Systems, Santa Barbara, USA*, 1990.

John A. Bateman. The theoretical status of ontologies in natural language processing. In *Proceedings of the workshop on Text Representation and Domain Modelling – Ideas from Linguistics and AI, Berlin, Germany*, 1991.

Camille Ben Achour. Guiding scenario authoring. In *Proceeding of the* 8^{th} *European-Japanese Conference on Information Modelling and Knowledge Bases, Finland*, 1998.

William J. Black. Acquisition of conceptual data models from natural language descriptions. In *Proceedings of the* 3^{rd} *European conference of the ACL, Copenhagen, Denmark*, pages 241–248, 1987.

Barry W. Boehm. A spiral model of software development and enhancement. In Richard H. Thayer, editor, *Software Engineering Project Management*, pages 128–142. IEEE Computer Society Press, 1987. Reprinted in Thayer and Dorfman (1997, p. 416–430).

Susan E. Brennan, Marilyn Walker Friedman, and Carl J. Pollard. A centering approach to pronouns. In *Proceedings of the* 25^{th} *Annual Meeting of the Association for Computational Linguistics, Stanford University, Stanford, California, USA*, pages 155–162, 1987.

Frederick P. Brooks. *The Mythical Man-Month.* Addison-Wesley, anniversary edition, 1995.

Janis Bubenko, Colette Rolland, P. Loucopoulos, and V. DeAntonellis. Facilitating fuzzy to formal requirements modelling. In *Proceedings of the* 1^{st} *IEEE International Conference on Requirements Engineering, Colorado-Springs, USA*, 1994.

J.F.M. Burg and R.P. van de Riet. The impact of linguistics on conceptual models: Consistency and understandability. In *Proceedings of the* 1^{st} *Workshop on Applications of Natural Language to Databases, Versailles, France*, 1995.

C-Star Consortium. *Dialogue act annotation*, 1998.

Denis Carcagno and Lidija Iordanskaja. Content determination and text structuring: two interrelated processes. In Helmut Horacek and Michael Zock, editors, *New concepts in natural language generation: Planning, realization, and systems*, chapter 1, pages 10–26. Pinter publishers, 1993.

Peter Pin-Shan Chen. The entity-relationship model – Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.

Peter Pin-Shan Chen. English sentence structure and entity-relationship diagrams. *Information Sciences*, 29:127–149, 1983.

Lawrence Chung, Brian A. Nixon, and Eric Yu. Using quality requirements to systematically develop quality software. In *Proceedings of the* 4^{th} *international conference on software quality, McLean, Virginia, USA*, 1994.

Herbert H. Clark and Susan E. Haviland. Comprehension and the given – New contract. In Roy O. Freedle, editor, *Discourse Production and Comprehension*, chapter 1, pages 1–40. Ablex publishing corporation, Norwood, New Jersey, USA, 1977.

Philip R. Cohen, Jerry Morgan, and Martha E. Pollack, editors. *Intentions in Communication.* The MIT Press, 1990.

Robin Cohen. A computational theory of the function of clue words in argument understanding. In *Proceedings of the* 22^{nd} *annual meeting of the Association for Computational Linguistics, Stanford University, California, USA*, pages 251–258, 1984.

Robin Cohen. Analyzing the structure of argumentative discourse. *Computational Linguistics*, 13(1–2):11–24, 1987.

Michael A. Covington. From English to Prolog via Discourse Representation Theory. Technical Report 01-0024, Advanced Computational Methods Center, University of Georgia, 1988.

Michael A. Covington. *Natural Language Processing for Prolog Programmers.* Prentice Hall, 1994.

Robert Dale. *Generating referring expressions: Constructing descriptions in a domain of objects and processes.* The MIT Press, 1992.

Anne Dardenne, Axel van Lamsweerde, and Stephen Fickas. Goal-directed requirements acquisition. *Science of Computer Programming*, 20:3–50, 1993.

Robert Darimont. *Process Support for Requirements Elicitation.* PhD thesis, Université catholique de Louvain, Louvain-la-Neuve, Belgium, 1995.

Robert Darimont and Axel van Lamsweerde. Formal refinement patterns for goal-driven requirements elaboration. In *Proceedings of the fourth ACM symposium on the foundation of software engineering, San Fransisco, USA*, 1996.

Alan Davis. A comparison of techniques for the specification of external system behavior. *Communications of the ACM*, 31(9):1098–1115, 1988.

Alan M. Davis. *Software Requirements: Objects, Functions, and States.* Prentice Hall, revised edition, 1993.

Wieland Eckert, Esther Levin, and Roberto Pieraccini. User modeling for spoken dialogue system evaluation. Technical Report 97.33.1, AT&T Labs Research, 1997.

Wieland Eckert, Esther Levin, and Roberto Pieraccini. Automatic evaluation of spoken dialogue systems. Technical Report 98.9.1, AT&T Labs Research, 1998.

Robin Fawcett, Eduard H. Hovy, David McDonald, Marie Meteer, Donia Scott, and Koenraad deSmedt, editors. *Proceedings of the 7^{th} International Workshop on Natural Language Generation, Kennebunkport, Maine, USA*, 1994.

Christiane Fellbaum. English verbs as a semantic net. *International Journal of Lexicography*, 3(4):278–301, 1990.

George Ferguson and James F. Allen. TRIPS: An intelligent integrated problem-solving assistant. In *Proceedings of the 15^{th} National Conference on Artificial Intelligence, Madison, Wisconsin, USA*, pages 26–30, 1998.

George Ferguson, James F. Allen, and Brad Miller. TRAINS-95: Towards a mixed-initiative planning assistant. In *Proceedings of the 3^{rd} conference on Artificial Intelligence Planning Systems, Edinburgh, Scotland*, pages 70–77, 1996.

Charles J. Fillmore. The case for case. In Emmon Bach and Robert T. Harms, editors, *Universals in Linguistics Theory*, pages 1–88. Holt, Rinehart and Winston, 1968.

Norbert E. Fuchs. Specification are (preferably) executable. *IEE/BCS Software Engineering Journal*, 7(2):323–334, 1992.

Norbert E. Fuchs, Uta Schwertel, and Rolf Schwitter. Attempto controlled English - Not just another logic specification language. In *Proceedings of the 8^{th} International Workshop on Logic-based Program Synthesis and Transformation, Manchester, UK*, 1998.

Norbert E. Fuchs and Rolf Schwitter. Attempto - controlled natural language for requirements specifications. In *Proceedings of the 7^{th} ILPS workshop on logic programming environments, Portland, Oregon, USA*, 1995.

Norbert E. Fuchs and Rolf Schwitter. Attempto controlled English (ACE). In *Proceedings of the 1^{st} international Workshop on Controlled Language Applications, Leuven, Belgium*, 1996.

L.T.F. Gamut. *Logic, language, and meaning*, volume 2. Intensional Logic and Logical Grammar. University of Chicago Press, 1991.

H. P. Grice. Logic and conversation. In Peter Cole and Jerry Morgan, editors, *Speech Acts*, volume 3 of *Syntax and Semantics*, pages 41–58. Academic Press, New York, 1975.

Derek Gross and Katherine J. Miller. Adjectives in wordnet. *International Journal of Lexicography*, 3(4):265–277, 1990.

Barbara J. Grosz. The representation and use of focus in dialogue understanding. Technical Report 151, SRI International, 1977.

Barbara J. Grosz. Focusing and description in natural language dialogues. In Aravind K. Joshi, Bonnie L. Webber, and Ivan A. Sag, editors, *Elements of discourse understanding*, chapter 3, pages 84–105. Cambridge University Press, 1981.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–226, 1995.

Barbara J. Grosz, Araving K. Joshi, and Scott Weinstein. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the* 21^{st} *annual meeting of the Association for Computational Linguistics, Cambridge, Massachusetts, USA*, pages 44–50, 1983.

Barbara J. Grosz and Karen E. Lochbaum. Intersegment relations and intentions. In Donia Scott and Eduard H. Hovy, editors, *NATO Workshop: Burning Issues in Discourse, Maratea, Italy*, pages 20–26, 1993.

Barbara J. Grosz and Candace L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

Barbara J. Grosz and Candace L. Sidner. Plans for discourse. In Cohen et al. (1990), chapter 20, pages 417–444.

Barbara J. Grosz and Candace L. Sidner. Lost intuitions and forgotten intentions. Technical Report 95-02, Lotus, 1995.

Raymonde Guindon. Anaphora resolution: Short-term memory and focusing. In *Proceedings of the* 23^{rd} *annual meeting of the Association for Computational Linguistics, Chicago, Illinois, USA*, pages 218–227, 1985.

Udo Hahn and Michael Strube. Centering-in-the-large: Computing referential discourse segments. In *Proceedings of the* 35^{th} *Annual Meeting of the Association for Computational Linguistics and the* 8^{th} *Conference of the European Chapter of the Association for Computational Linguistics, Madrid, Spain*, pages 104–111. Association for Computational Linguistics, 1997.

Steve Hanks, Martha E. Pollack, and Paul R. Cohen. Benchmarks, testbeds, controlled experimentation, and the design of agent architectures. *AI Magazine*, 14(4):17–42, 1993.

J.-P. Haton, N. Bouzid, F. Charpillet, M.-Ch. Haton, B. Lâasri, H. Lâasri, P. Marquis, T. Mondot, and A. Napoli. *Le Raisonnement en Intelligence Artificielle*. InterEditions, Paris, France, 1991.

Robert Helm and Stephen Fickas. Scare tactics: Evaluating problem decompositions using failure scenarios. Technical Report CIS-TR-92-06, University of Oregon, 1992.

Janet Hitzeman and Massimo Poesio. Long distance pronominalisation and global focus. In *Proceedings of the* 17^{th} *International Conference on Computational Linguistics and* 36^{th} *annual conference of the Association for Computational Linguistics, Montréal, Canada*, pages 550–556, 1998.

Eduard H. Hovy. *Generating natural language under pragmatic constraints.* Laurence Erlbaum Associates, 1988a.

Eduard H. Hovy. Planning coherent multisentential text. In *Proceedings of the* 26^{th} *annual conference of the Association for Computational Linguistics, Buffalo, New York, USA*, pages 163–169, 1988b.

Eduard H. Hovy. Approaches to the planning of coherent text. In Paris et al. (1991), chapter 3, pages 83–102.

Eduard H. Hovy. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63:341–385, 1993.

Eduard H. Hovy and Kathleen F. McCoy. Focusing your RST: A step toward generating coherent multisentential text. In *Proceedings of the* 11^{th} *annual conference of the Cognitive Science Society, Ann Arbor, Michigan, USA*, pages 667–674, 1989.

Pei Hsia, Jayarajan Samuel, Jerry Gao, David Kung, Yasufumi Toyoshima, and Chris Chen. Formal approach to scenario analysis. *IEEE Software*, 11(2):33–41, 1994.

Xiaorong Huang. Planning argumentative texts. In *Proceedings of the* 15^{th} *International Conference on Computational Linguistics, Kyoto, Japan*, pages 329–333, 1994a.

Xiaorong Huang. Planning reference choices for argumentative texts. In Fawcett et al. (1994), pages 145–152.

Xiaorong Huang and Armin Fiedler. Proof presentation as an application of NLG. In Pollack (1997), pages 965–970.

IEEE. *Proceedings of the* 1^{st} *IEEE international symposium on Requirements Engineering, Los Alamitos, USA*. IEEE Computer Society Press, 1993.

Yasunori Ishihara, Hiroyuki Seki, and Tadao Kasami. A translation method from natural language specifications into formal specifications using contextual dependencies. In *Proceedings of the* 1^{st} *IEEE international symposium on Requirements Engineering, Los Alamitos, USA* IEE (1993), pages 232–239.

Masato Ishizaki. *Mixed-Initiative Natural Language Dialogue with Variable Communicative Modes.* PhD thesis, University of Edinburgh, Scotland, 1997.

Arne Jönsson and Nils Dahlbäck. Proceedings of the 5^{th} european conference on speech technology and communication, rhodes, greece. In Kokkinakis et al. (1997), pages 2215–2218.

Hans Kamp. A theory of truth and semantic representation. In Jeroen A.G. Groenengijk, Theo M.V. Janssen, and Martin B.J. Stokhof, editors, *Formal Methods in the Study of Language*, pages 277–322. Mathematical Centre, Amsterdam, Netherlands, 1981.

Joachim Karlsson. Software requirements prioritizing. In *Proceedings of the 2^{nd} IEEE International Conference on Requirements Engineering, Colorado-Springs, CO, USA*, pages 110–118, 1996.

Andrew Kehler. Current theories of centering for pronoun interpretation: A critical evaluation. *Computational Linguistics*, 23(3):467–475, 1997.

Richard Kittredge, Tanya Korelsky, and Owen Rambow. On the need for domain communication knowledge. *Computational Intelligence*, 7(4):305–314, 1991.

Alistair Knott and Robert Dale. Choosing a set of coherence relations for text-generation: A data-driven approach. In Giovanni Adorni and Michael Zock, editors, *Trends in natural language generation: An artificial intelligence perspective*, number 1036 in Lecture Notes in Artificial Intelligence, pages 47–67. Springer-Verlag, 1996.

G. Kokkinakis, N. Fakotakis, and E. Dermatas, editors. *Proceedings of the 5^{th} European Conference on Speech Technology and Communication, Rhodes, Greece.* TYPOFFSET, Patras, Greece, 1997.

Robert Kowalski. *Logic for problem solving.* Elsevier Science Publishing Company, 1979.

Robert Kowalski and Marek Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 1986.

Yannis Labrou. *Semantics for an agent communication language.* PhD thesis, University of Maryland, USA, 1996.

Yannis Labrou and Tim Finin. Semantics and conversations for an agent communication language. In Pollack (1997), pages 584–591.

Renaud Lecœuche, Chris Mellish, Catherine Barry, and Dave Robertson. User-system dialogues and the notion of focus. *The Knowledge Engineering Review*, 4(13):381–408, 1998a.

Renaud Lecœuche, Chris Mellish, and Dave Robertson. A framework for requirements elicitation through mixed-initiative dialogue. In *Proceedings of the 3^{rd} International Conference on Requirements Engineering, Colorado-Springs, CO, USA*, pages 190–196. IEEE, IEEE Computer Society Press, 1998b.

Renaud Lecœuche, Dave Robertson, and Catherine Barry. Using focus rules in requirements elicitation dialogues. In Thomas Dean, editor, *Proceedings of the 16^{th} International Joint Conference on Artificial Intelligence, Stockholm, Sweden*, pages 649–654. Morgan Kaufman, 1999.

Julio Cesar Sampaio do Prado Leite. Eliciting requirements using a natural language based approach: The case of the meeting scheduler problem. Technical Report 13/93, Pontifícia Universidade Católica do Rio de Janeiro, 1993.

Julio Cesar Sampaio do Prado Leite and Ana Paula M. Franco. A strategy for conceptual model acquisition. In *Proceedings of the* 1^{st} *IEEE international symposium on Requirements Engineering, Los Alamitos, USA* IEE (1993), pages 243–246.

Willem J.M. Levelt. *Speaking: From intention to articulation.* The MIT Press, 1989.

Charlotte Linde. Focus of attention and the choice of pronouns in discourse. In Talmy Givón, editor, *Discourse and Syntax*, volume 12 of *Syntax and Semantics*, pages 337–354. Academic Press, New York, 1979.

Charlotte Linde and Joseph A. Goguen. Structure of planning discourse. *Journal of social biological structures*, 1:219–251, 1978.

Gitte Lindgaard. *Usability testing and system evaluation.* Chapman & Hall, 1994.

Susann Luperfoy. The representation of multimodal user interface dialogues using discourse pegs. In *Proceedings of the* 30^{th} *Annual Meeting of the Association for Computational Linguistics, Newark, Delaware, USA*, pages 22–31, 1992.

Benjamin Macias and Stephen Pulman. Natural language processing for requirements specifications. In Felix Remill and Tom Anderson, editors, *Safety-critical Systems*, chapter 4, pages 67–89. Chapman & Hall, 1993.

William C. Mann and Sandra A. Thompson. Rhetorical Structure Theory: A theory of text organization. Technical Report RS-87-190, University of Southern California, Information Science Institute, 1987.

Kathleen F. McCoy. Reasoning on a highlighted user model to respond to misconceptions. *Computational Linguistics*, 14(3):52–63, 1988.

Kathleen F. McCoy and Jeannette Cheng. Focus of attention: Constraining what can be said next. In Paris et al. (1991), chapter 4, pages 103–124.

Kathleen R. McKeown. Discourse strategies for generating natural-language text. *Artificial Intelligence*, 27(1):1–41, 1985a.

Kathleen R. McKeown. *Text generation – Using discourse strategies and focus constraints to generate natural language text.* Studies in Natural Language Processing. Cambridge University Press, 1985b.

Bertrand Meyer. On formalism in specifications. *IEEE Software*, 2(1):6–26, 1985.

Luisa Mich. NL-OOPS: From natural language to object oriented requirements using the natural language processing system LOLITA. *Natural Language Engineering*, 2(2):161–187, 1996.

George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97, 1956.

George A. Miller. Nouns in wordnet: A lexical inheritance system. *International Journal of Lexicography*, 3(4):245–264, 1990.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.

Vibhu O. Mittal, Johanna D. Moore, Giuseppe Carenini, and Steven Roth. Describing complex charts in natural language: A caption generation system. *Computational Linguistics*, 24(3):431–467, 1998.

Jens-Uwe Moeller. Domain-related focus-shifting constraints in dialogues with knowledge based systems. In Giovanni Adorni and Michael Zock, editors, *Trends in natural language generation: An artificial intelligence perspective*, number 1036 in Lecture Notes in Artificial Intelligence, pages 188–204. Springer-Verlag, 1996.

Johanna D. Moore and Cécile L. Paris. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–694, 1993.

Johanna D. Moore and Martha E. Pollack. A problem for RST: The need for multi-level discourse analysis. *Computational Linguistics*, 18(4):537–544, 1992.

Johanna D. Moore and William R. Swartout. A reactive approach to explanation. In N. S. Sridharan, editor, *Proceedings of the 11^{th} International Joint Conference on Artificial Intelligence, Detroit, MI, USA*, pages 1504–1510. Morgan Kaufmann, 1989.

Ana M. Moreno. Object-oriented analysis from textual specifications. In *Proceedings of the 9^{th} international conference on Software Engineering and Knowledge Engineering, Madrid, Spain*, 1997.

Megan G. Moser and Johanna D. Moore. Toward a synthesis of two accounts of discourse structure. *Computational Linguistics*, 22(3):409–420, 1996.

John Mylopoulos, Laurence Chung, and Brian Nixon. Representing and using nonfunctional requirements: A process-oriented approach. *IEEE transaction on software engineering*, 18(6):483–497, 1992.

Sastry Nanduri and Spencer Rugaber. Requirements validation via automated language parsing. *Journal of Management Information Systems*, 12(2):9–19, 1996.

Cécile L. Paris, William R. Swartout, and William C. Mann, editors. *Natural Language Generation in Artificial Intelligence and Computational Linguistics.* Kluwer Academic Publishers, 1991.

Derek Partridge. *Engineering Artificial Intelligence Software.* Intellect Books, Oxford, United Kingdom, 1992.

Livia Polanyi. A theory of discourse structure and discourse coherence. In W. H. Eilfort, P. D. Kroeber, and K.L Peterson, editors, *Papers from the general session of the 21^{st} regional meeting of the Chicago linguistics society, Chicago, Illinois, USA*, pages 306–322, 1985.

Martha E. Pollack, editor. *Proceedings of the 15^{th} International Joint Conference on Artificial Intelligence, Nagoya, Japan.* Morgan Kaufman, 1997.

Chris A. Reed and Derek P. Long. Content ordering in the generation of persuasive discourse. In Pollack (1997), pages 1022–1027.

Chris A. Reed and Derek P. Long. Ordering and focusing in an architecture for persuasive discourse planning. In W. Hoeppner, editor, *Proceedings of the* 6^{th} *European Workshop on Natural Language Generation, Duisburg, Germany*, 1997b.

Chris A. Reed, Derek P. Long, and Maria Fox. Context and focusing in argumentative dialogue planning. In *Proceedings of the International Conference on Context, Rio de Janeiro, Brazil*, pages 88–99, 1997.

Rachel Reichman. Conversational coherency. *Cognitive Science*, 2:283–327, 1978.

Rachel Reichman. Plain speaking: A theory and grammar of spontaneous discourse. Technical Report 4681, Bolt, Beranek and Newman Inc., Cambridge, MA, USA, 1981.

Rachel Reichman. Extended person-machine interface. *Artificial Intelligence*, 22:157–218, 1984.

Rachel Reichman. *Getting Computers to Talk Like You and Me.* The MIT Press, Cambridge, MA, USA, 1985.

Ehud Reiter. Has a conscensus NL generation architecture appeared, and is it psycholinguistically plausible? In Fawcett et al. (1994), pages 163–170.

Ehud Reiter. NLG vs. templates. In *Proceedings of the* 5^{th} *European workshop on natural language generation, Leiden, The Netherlands*, 1995.

Howard B. Reubenstein. Automated acquisition of evolving informal descriptions. Technical Report TR-1205, MIT Artificial Intelligence Laboratory, 1990.

Charles Rich and Candace L. Sidner. Collagen: When agents collaborate with people. In *Proceedings of the* 1^{st} *Conference on Autonomous Agents, Marina del Rey, California, USA*, 1997.

David Stuart Robertson, Mike Uschold, Alan Bundy, and Robert Muetzelfeldt. The ECO program construction system: Ways of increasing its representational power and their effects on the user interface. *International Journal of Man Machine Studies*, 31(1):1–26, 1989.

Colette Rolland. A contextual approach for the requirements engineering process. In *Proceedings of the* 6^{th} *international conference on Software Engineering and Knowledge Engineering, Vilnius, Lithuenia*, 1994.

Colette Rolland and Camille Ben Achour. Guiding the construction of textual use case specifications. *Data and knowledge engineering journal*, 25(1–2):125–160, 1998.

Colette Rolland and C. Proix. A natural language approach for requirements engineering. In P. Loucopoulos and R. Zicari, editors, *Conceptual modelling, databases and CASE: An integrated view of the information systems development.* John Wiley & Sons, 1992.

Thomas L. Saaty. How to make a decision: The Analytic Hierarchy Process. *European Journal of Operational Research*, 48:9–26, 1990.

Roger C. Schank. Rules and topics in conversation. *Cognitive Science*, 1:421–441, 1977.

Rolf Schwitter and Norbert Fuchs. Attempto - from specifications in controlled natural language towards executable specifications. In *Proceedings of the GI EMISA Workshop, Tutzing, Germany*, 1996.

John R. Searle. *Expression and meaning*. Cambridge University Press, 1979.

Samira Si-Saïd and Colette Rolland. Formalising guidance for the CREWS goal-scenario approach to requirements engineering. In *Proceedings of the* 8^{th} *European-Japanese Conference on Information Modelling and Knowledge Bases, Vamala, Finland*, 1998.

Penelope Sibun. Generating text without trees. *Computational Intelligence*, 8(1):102–122, 1992.

Candace L. Sidner. Towards a computational theory of definite anaphora comprehension in English discourse. Technical Report TR-357, MIT Artificial Intelligence Laboratory, 1979.

Candace L. Sidner. Focusing and discourse. *Discourse Processes*, 6:107–130, 1983.

Teresa Sikorski and James F. Allen. A task-based evaluation of the TRAINS-95 dialogue system. In Elisabeth Maier, Marion Mast, and Susann LuperFoy, editors, *Proceedings of the ECAI Workshop on Dialogue Processing in Spoken Languages Systems, Budapest, Hungary*, volume 1236 of *LNAI*, pages 207–220, Berlin, 1996. Springer.

Ronnie W. Smith. Effective spoken natural language dialog requires variable initiative behavior: An empirical study. In *Proceedings of the AAAI fall symposium on human-computer collaboration: Reconciling theory, synthesizing practice*, pages 101–106, 1993.

Ronnie W. Smith. Initiative-dependent features of human-computer dialogs in a task-assistance domain. In *Proceedings of the Energy Information Management Conference*, pages 41–48, 1996.

Ronnie W. Smith. Practical issues in mixed-initiative natural language dialog: An experimental perspective. In *Proceedings of the 1997 AAAI Spring Symposium on Computational Models for Mixed-Initiative Interaction, Stanford, California, USA*, pages 158–162, 1997.

Ronnie W. Smith, D. Richard Hipp, and Alan W. Biermann. A dialog control algorithm and its performance. In *Proceedings of the* 3^{rd} *conference on Applied Natural Language Processing*, pages 9–16, 1992.

Ian Sommerville. *Software Engineering*. Addison-Wesley, 5^{th} edition, 1995.

Ian Sommerville and Pete Sawyer. *Requirements Engineering: A good practice guide*. John Wiley & Sons, 1997.

Karen Sparck Jones and Julia R. Galliers. *Evaluating Natural Language Processing Systems*, volume 1083 of *Lecture Notes in Artificial Intelligence*. Springer, 1996.

Michael Strube. Never look back: An alternative to centering. In *Proceedings of the* 17^{th} *International Conference on Computational Linguistics and* 36^{th} *annual conference of the Association for Computational Linguistics, Montréal, Canada*, pages 1251–1257, 1998.

Michael Strube and Udo Hahn. Functional centering. In *Proceedings of the 34^{th} Annual Meeting of the Association for Computational Linguistics, Santa Cruz, California, USA*, pages 270–277, 1996.

William R. Swartout and Stephen W. Smoliar. Explanation: A source of guidance for knowledge representation. In *Proceedings of the Workshop on Knowledge Representation and Organization in Machine Learning, Eringerfeld, Germany*, 1987.

Richard Thayer and Martin Dorfman, editors. *Software Requirements Engineering.* IEEE Computer Society Press, 2^{nd} edition, 1997.

Andre Thayse, editor. *From modal logic to deductive databases : Introducing a logic based approach to artificial intelligence.* John Wiley & Sons, 1989.

David R. Traum, James F. Allen, George Ferguson, Peter A. Heeman, Chung Hee Hwang, Tsuneaki Kato, Nathaniel Martin, Massimo Poesio, and Lenhart K. Schubert. Integrating natural language understanding and plan reasoning in the TRAINS-93 conversation system. In *Proceedings of the AAAI Spring Symposium on Active NLP*, 1994.

Marilyn A. Walker. *Informational Redundancy and Resource Bounds in Dialogue.* PhD thesis, University of Pennsylvania, 1993.

Marilyn A. Walker. The effect of resource limits and task complexity on collaborative planning in dialogue. *Artificial Intelligence*, 85(1–2):181–243, 1996a.

Marilyn A. Walker. Limited attention and discourse structure. *Computational Linguistics*, 22(2):255–264, 1996b.

Marilyn A. Walker. Centering, anaphora resolution, and discourse structure. In Walker et al. (1997b).

Marilyn A. Walker, Jeanne Fromer, Giuseppe Di Fabbrizio, Craig Mestel, and Don Hindle. What can I say?:Evaluating a spoken language interface to email. In *Proceedings of the conference on Human Factors in Computing Systems, Los Angeles, California, USA*, pages 582–589, 1998.

Marilyn A. Walker, Don Hindle, Jeanne Fromer, Giuseppe Di Fabbrizio, and Craig Mestel. Evaluating competing agent strategies for a voice email agent. In Kokkinakis et al. (1997), pages 2219–2222.

Marilyn A. Walker, Masayo Iida, and Sharon Cote. Centering in japanese discourse. In *Proceedings of the 13^{th} International Conference on Computational Linguistics, Helsinki, Finland*, 1990.

Marilyn A. Walker, Aravind K. Joshi, and Ellen F. Prince, editors. *Centering in discourse.* Oxford University Press, 1997b.

Marilyn A. Walker, Aravind K. Joshi, and Ellen F. Prince. Centering in naturally-occurring discourse: An overview. In *Centering in discourse* Walker et al. (1997b).

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. PARADISE: A framework for evaluating spoken dialogue agents. In *Proceedings of the 35^{th} annual meeting of the Association for Computational Linguistics, Madrid, Spain*, 1997d.

Paul T. Ward. The transformation schema: An extension of the data flow diagram to represent control and timing. *IEEE Transactions on Software Engineering*, 12(2):198–210, 1986.

Roel J. Wieringa. *Requirements Engineering – Frameworks for understanding.* John Wiley & Sons, 1996.

R. Michael Young and Johanna D. Moore. DPOCL: A principled approach to discourse planning. In Fawcett et al. (1994), pages 13–20.

R. Michael Young, Johanna D. Moore, and Martha E. Pollack. Towards a principled representation of discourse plans. In *Proceedings of the* 16^{th} *Conference of the Cognitive Science Society , Atlanta, GA, USA*, pages 946–951, 1994.

Eric Yu. *Modelling Strategic Relationships For Process Reengineering.* PhD thesis, University of Toronto, 1994. also available as report DKBS-TR-94-6.

Appendix

Paul T. Ward. The transformation schema: An extension of the data flow diagram to represent control and timing. *IEEE Transactions on Software Engineering*, 12(2):198–210, 1986.

Roel J. Wieringa. *Requirements Engineering: Frameworks for understanding*. John Wiley & Sons, 1996.

R. Michael Young and Johanna D. Moore. DPOCL: A principled approach to discourse planning. In Fawcett et al. [1994], pages 13–20.

R. Michael Young, Johanna D. Moore, and Martha E. Pollack. Towards a principled representation of discourse plans. In *Proceedings of the 16th Conference of the Cognitive Science Society*, Atlanta, GA, 1994, pages 946–951, 1994.

Eric Yu. *Modelling Strategic Relationships for Process Reengineering*. PhD thesis, University of Toronto, 1995. Also Technical Report DKBS-TR-94-6.

Appendix A

Entity-Relationship Models

A.1 Short WWW site design Entity-Relationship model

entityset(research group)	entityset(publication set)
entityset(researcher set)	entityset(site)
entityset(home page)	entityset(page)
relationset(present)	relationset(involve)
relationset(describe)	relationset(link)
cardmin(research group, 1)	cardmax(research group, 1)
cardmin(publication set, 1)	cardmax(publication set, 1)
cardmin(researcher set, 1)	cardmax(researcher set, 1)
cardmin(site, 0)	cardmax(site, ∞)
cardmin(home page, 0)	cardmax(home page, ∞)
cardmin(page, 0)	cardmax(page, ∞)
role(present, presented by)	role(present, presenting)
role(involve, involved in)	role(involve, involving)
role(describe, described by)	role(describe, describing)
role(link, linked from)	role(link, linked to)
rolefiller(present, presented by, site)	
rolefiller(present, presented by, page)	
rolefiller(present, presenting, research group)	
rolefiller(present, presenting, researcher set)	
rolefiller(present, presenting, publication set)	
rolefiller(involve, involved in, research group)	
rolefiller(involve, involving, publication set)	
rolefiller(involve, involving, researcher set)	
rolefiller(describe, described by, home page)	
rolefiller(describe, describing, site)	
rolefiller(link, linked from, home page)	
rolefiller(link, linked from, page)	
rolefiller(link, linked to, home page)	

rolefiller(link, linked to, page)
cardmin(present, presented by, 0)
cardmax(present, presented by, 1)
cardmin(present, presenting, 0)
cardmax(present, presenting, 1)
cardmin(involve, involved in, 1)
cardmax(involve, involved in, 1)
cardmin(involve, involving, 1)
cardmax(involve, involving, 1)
cardmin(describe, described by, 0)
cardmax(describe, described by, 1)
cardmin(describe, describing, 1)
cardmax(describe, describing, 1)
cardmin(link, linked from, 0)
cardmax(link, linked from, ∞)
cardmin(link, linked to, 0)
cardmax(link, linked to, ∞)
rolecardmin(present,presenting, 1)
attribute(home page, title, text)
attribute(page, title, text)

A.2 Full WWW site design Entity-Relationship model

entityset(research group)	
entityset(researcher set)	entityset(researcher)
entityset(publication set)	entityset(publication)
entityset(event set)	entityset(event)
entityset(link set)	entityset(link)
entityset(page)	entityset(site)
entityset(home page)	entityset(navigator)
entityset(event type)	entityset(useful link)
relationset(involve)	relationset(belong-rs)
relationset(belong-es)	relationset(belong-pubs)
relationset(belong-ls)	relationset(present-site)
relationset(present)	relationset(link)
relationset(homepage)	relationset(contain)
relationset(display)	relationset(event type)
relationset(author)	relationset(speaker)
relationset(peo)	relationset(go)
cardmin(research group, 1)	cardmax(research group, 1)
cardmin(researcher set, 1)	cardmax(researcher set, 1)
cardmin(publication set, 1)	cardmax(publication set, 1)
cardmin(event set, 1)	cardmax(event set, 1)
cardmin(link set, 1)	cardmax(link set, 1)
cardmin(site, 1)	cardmax(site, 1)
cardmin(navigator, 0)	cardmax(navigator, 1)

cardmin(researcher, 0)	cardmax(researcher, ∞)
cardmin(publication, 0)	cardmax(publication, ∞)
cardmin(event, 0)	cardmax(event, ∞)
cardmin(event type, 0)	cardmax(event type, ∞)
cardmin(home page, 0)	cardmax(home page, ∞)
cardmin(useful link, 0)	cardmax(useful link, ∞)
role(involve, involving)	role(involve, involved in)
role(belong-rs, belonging to)	role(belong-rs, containing)
role(belong-es, belonging to)	role(belong-es, containing)
role(belong-pubs, belonging to)	role(belong-pubs, containing)
role(belong-ls, belonging to)	role(belong-ls, containing)
role(present-site, presented by)	role(present-site, presenting)
role(present, presented by)	role(present, presenting)
role(link, linked from)	role(link, linked to)
role(homepage, describing)	role(homepage, described by)
role(contain, pointing to)	role(contain, pointed by)
role(display, displaying)	role(display, displayed by)
role(event type, of type)	role(event type, containing)
role(author, written by)	role(author, writing)
role(speaker, presented by)	role(speaker, presenting)
role(peo, having public event organised by)	role(peo, organising public events of)
role(go, organised by)	role(go, organising)

```
rolefiller(involve, involved in, research group)
rolefiller(involve, involving, researcher set)
rolefiller(involve, involving, publication set)
rolefiller(involve, involving, event set)
rolefiller(involve, involving, link set)
rolefiller(belong-rs, belonging to, researcher set)
rolefiller(belong-rs, containing, researcher)
rolefiller(belong-es, belonging to, event set)
rolefiller(belong-es, containing, event)
rolefiller(belong-pubs, belonging to, publication set)
rolefiller(belong-pubs, containing, publication)
rolefiller(belong-ls, belonging to, link set)
rolefiller(belong-ls, containing, useful link)
rolefiller(present site, presented by, site)
rolefiller(present site, presenting, research group)
rolefiller(present, presented by, page)
rolefiller(present, presenting, researcher set)
rolefiller(present, presenting, publication set)
rolefiller(present, presenting, event set)
rolefiller(present, presenting, link set)
rolefiller(link, linked to, page)
rolefiller(link, linked to, homepage)
rolefiller(link, linked from, page)
rolefiller(link, linked from, homepage)
```

```
rolefiller(homepage, describing, site)
rolefiller(homepage, described by, homepage)
rolefiller(contain, pointing to, page)
rolefiller(contain, pointing to, homepage)
rolefiller(contain, pointed by, navigator)
rolefiller(display, displaying, navigator)
rolefiller(display, displayed by, page)
rolefiller(display, displayed by, homepage)
rolefiller(event type, of type, event type)
rolefiller(event type, containing, event)
rolefiller(author, written by, researcher)
rolefiller(author, writing, publication)
rolefiller(speaker, presented by, researcher)
rolefiller(speaker, presenting, event)
rolefiller(peo, having public event organised by, researcher)
rolefiller(peo, organising public events of, research group)
rolefiller(go, organised by, researcher)
rolefiller(go, organising, research group)
```

```
cardmin(involve, involving, 1)
cardmax(involve, involving, 1)
cardmin(involve, involved in, 1)
cardmax(involve, involved in, 1)
cardmin(belong-rs, belonging to, 1)
cardmax(belong-rs, belonging to, 1)
cardmin(belong-rs, containing, 0)
cardmax(belong-rs, containing, ∞)
cardmin(belong-es, belonging to, 1)
cardmax(belong-es, belonging to, 1)
cardmin(belong-es, containing, 0)
cardmax(belong-es, containing, ∞)
cardmin(belong-pubs, belonging to, 1)
cardmax(belong-pubs, belonging to, 1)
cardmin(belong-pubs, containing, 0)
cardmax(belong-pubs, containing, ∞)
cardmin(belong-ls, belonging to, 1)
cardmax(belong-ls, belonging to, 1)
cardmin(belong-ls, containing, 0)
cardmax(belong-ls, containing, ∞)
cardmin(present site, presented by, 0)
cardmax(present site, presented by, 1)
cardmin(present site, presenting, 1)
cardmax(present site, presenting, 1)
cardmin(present, presented by, 0)
cardmax(present, presented by, 1)
cardmin(present, presenting, 1)
cardmax(present, presenting, 1)
cardmin(link, linked to, 0)
```

cardmax(link, linked to, ∞)	
cardmin(link, linked from, 0)	
cardmax(link, linked from, ∞)	
cardmin(homepage, describing, 1)	
cardmax(homepage, describing, 1)	
cardmin(homepage, described by, 0)	
cardmax(homepage, described by, 1)	
cardmin(display, displaying, 0)	
cardmax(display, displaying, 1)	
cardmin(display, displayed by, 1)	
cardmax(display, displayed by, ∞)	
cardmin(contain, pointing to, 1)	
cardmax(contain, pointing to, ∞)	
cardmin(contain, pointed by, 0)	
cardmax(contain, pointed by, 1)	
cardmin(event type, of type, 1)	
cardmax(event type, of type, 1)	
cardmin(event type, containing, 0)	
cardmax(event type, containing, ∞)	
cardmin(author, written by, 1)	
cardmax(author, written by, ∞)	
cardmin(author, writing, 0)	
cardmax(author, writing, ∞)	
cardmin(speaker, presented by, 1)	
cardmax(speaker, presented by, 1)	
cardmin(speaker, presenting, 0)	
cardmax(speaker, presenting, ∞)	
cardmin(peo, having public event organised by, 1)	
cardmax(peo, having public event organised by, 1)	
cardmin(peo, organising public events of, 1)	
cardmax(peo, organising public events of, 0)	
cardmin(go, organised by, 1)	
cardmax(go, organised by, 1)	
cardmin(go, organising, 0)	
cardmax(go, organising, 1)	
rolecardmax(present, presented by, 1)	rolecardmax(present, presenting, 1)
attribute(page, title)	attribute(homepage, title)
attribute(research group, name)	
attribute(research group, postal address)	
attribute(research group, phone number)	
attribute(research group, fax number)	
attribute(research group, mail address)	
attribute(researcher, name)	
attribute(researcher, www page address)	
attribute(researcher, status)	
attribute(event, day)	attribute(event, year)

attribute(event, time)
attribute(event, location)
attribute(event, title)
attribute(event, abstract)
attribute(event type, description)
attribute(event type, period)
attribute(publication, title)
attribute(publication, year)
attribute(publication, url)
attribute(publication, place of publication)
attribute(publication, abstract)
attribute(useful link, type)
attribute(useful link, url)
attribute(useful link, name)

A.3 Library Entity-Relationship model

entityset(department)	entityset(document)
entityset(title)	entityset(member)
cardmin(document,1)	
relationset(loss)	relationset(loan)
relationset(document reservation)	relationset(title reservation)
relationset(owner)	relationset(entitled)
role(loss, lost by)	role(loss, to report the loss of)
role(loan, borrowing)	role(loan, borrowed by)
role(document reservation, to reserve)	role(document reservation, reserved by)
role(title reservation, to reserve)	role(title reservation, reserved by)
role(owner, own by)	role(owner, possessing)
role(entitled, having)	role(entitled, to refer to a)
rolefiller(loss, lost by, member)	
rolefiller(loss, to report the loss of, document)	
rolefiller(loan, borrowing, document)	
rolefiller(loan, borrowed by, member)	
rolefiller(document reservation, to reserve, document)	
rolefiller(document reservation, reserved by, member)	
rolefiller(title reservation, to reserve, title)	
rolefiller(title reservation, reserved by, member)	
rolefiller(owner, own by, department)	
rolefiller(owner, possessing, document)	
rolefiller(entitled, having, title)	
rolefiller(entitled, to refer to a, document)	
cardmax(loss, lost by, 1)	cardmin(loss, lost by, 0)
cardmax(loss,to report the loss of, ∞)	cardmin(loss,to report the loss of, 0)
cardmax(loan, borrowing, ∞)	cardmin(loan, borrowing, 0)

cardmax(loan, borrowed by, 1)	cardmin(loan, borrowed by, 0)
cardmax(document reservation, to reserve, ∞)	cardmin(document reservation, to reserve, 0)
cardmax(document reservation, reserved by, ∞)	cardmin(document reservation, reserved by, 0)
cardmax(title reservation, to reserve, ∞)	cardmin(title reservation, to reserve, 0)
cardmax(title reservation, reserved by, ∞)	cardmin(title reservation, reserved by, 0)
cardmax(owner, own by, 1)	cardmin(owner, own by, 1)
cardmax(owner, possessing, ∞)	cardmin(owner, possessing, 0)
cardmax(entitled, having, 1)	cardmin(entitled, having, 1)
cardmax(entitled, to refer to a, ∞)	cardmin(entitled, referring to, 0)
attribute(document, status)	
attribute(title, name)	
attribute(title, author)	
attribute(member, name)	
attribute(member, address)	
attribute(loss, date)	
attribute(loss, amount of fine)	
attribute(title reservation, date)	
attribute(document reservation, date)	
attribute(loan, return date)	
attribute(loan, date of borrowing)	

A.4 Specification model Entity-Relationship model

entityset(entity set)	entityset(relation set)
entityset(role)	entityset(attribute)
cardmin(entity set,1)	
relationset(rolefiller)	relationset(role)
relationset(attribute)	
role(rolefiller,filling)	role(rolefiller, filled by)
role(role,having)	role(role, a role for)
role(attribute,having)	role(attribute, attribute of)
rolefiller(rolefiller, filling, role)	
rolefiller(rolefiller, filled by, entity set)	
rolefiller(role, having, role)	
rolefiller(role, a role for, relation set)	
rolefiller(attribute, having, attribute)	
rolefiller(attribute, attribute of, entity set)	
cardmax(rolefiller, filling, ∞)	cardmin(rolefiller, filling, 0)
cardmax(rolefiller, filled by, ∞)	cardmin(rolefiller, filled by, 1)
cardmax(role, having, 2)	cardmin(role, having, 2)

cardmax(role, a role for, 1)	cardmin(role,a role for, 1)
cardmax(attribute, having, ∞)	cardmin(attribute, having, 0)
cardmax(attribute, attribute of, 1)	cardmin(attribute, attribute of, 1)
attribute(entity set, cardmin)	
attribute(entity set, cardmax)	
attribute(entity set, name)	
attribute(role, cardmin)	
attribute(role, cardmax)	
attribute(role, overall cardmin)	
attribute(role, overall cardmax)	
attribute(role, name)	
attribute(relation set, name)	
attribute(attribute, name)	

Appendix B

Dialogue example

B.1 Worked example dialogue

```
research_group 1 is the research_group
Do you want a site presenting it ?
>> yes

A site presents the research_group
Which site presents the research_group ?
>> a new site

site 19 presents the research_group
Do you want a homepage describing site 19 ?
>> site 19 is described_by a new homepage
Analysing..........................................

homepage 21 describes site 19
What is the title of homepage 21 ?
>> "Research group"

Do you want a page linked_from it ?
>> yes

A page is linked_from Research group homepage 21
Which page is linked_from Research group homepage 21 ?
>> a new page

page 28 is linked_from Research group homepage 21
What is the title of page 28 ?
>> I want to speak_about the publication_set
Analysing.......................

publication_set 11 is the publication_set
```

```
it is involved_in the research_group
Do you want a site presenting it ?
```

B.2 Automated user's dialogue

S: research group 1 is the research group.
Do you want a site presenting it?
U: Yes.
S: A site presents the research group.
Which site presents the research group?
U: A new site.
S: Site 22 presents the research group.
Do you want a home page describing site 22?
U: Yes.
S: A home page describes site 22.
Which home page describes site 22?
U: A new home page.
S: Home page 24 describes site 22.
What is the title of home page 24?
U: research group.
S: Do you want it a page linked from it?
U: Yes.
S: A page is linked from research group home page 24.
Which page is linked from research group home page 24?
U: A new page.
S: Page 31 is linked from research group home page 24.
What is the title of page 31?
U: researcher set.
S: Do you want another home page linked to it?
U: No.
S: Do you want another page linked from it?
U: No.
S: Do you want the researcher set presented by it?
U: Yes.
S: The researcher set is presented by researcher set page 31.
researcher set 7 is the researcher set.
It is involved in the research group.
Do you want a home page linked from researcher set page 31?
U: I'll answer this question later.
I pass this question.
S: Do you want a page linked to researcher set page 31?
U: I'll answer this question later.
I pass this question.
S: Do you want a navigator pointing to it?
U: I'll answer this question later.
I pass this question.

S: Do you want a navigator displayed by it?
U: I'll answer this question later.
I pass this question.
S: Let's come back to our previous topic.
The project set is involved in the research group.
Project set 15 is the project set.
Do you want a site presenting it?
U: No.
S: Do you want a page presenting it?
U: Yes.
S: A page presents the project set.
Which page presents the project set?
U: A new page.
S: Page 43 presents the project set.
What is the title of page 43?
U: project set.
S: Do you want a page linked from it?
U: No.
S: Do you want a page linked to it?
U: No.
S: Do you want a home page linked from it?
U: No.
S: Do you want a home page linked to it?
U: Yes.
S: A home page is linked to project set page 43.
Which home page is linked to project set page 43?
U: home page 24.
S: Do you want project set page 43 linked from another home page?
U: I'll answer this question later.
I pass this question.
S: Do you want project set page 43 pointed by a navigator?
U: I'll answer this question later.
I pass this question.
S: Do you want it displaying a navigator?
U: I'll answer this question later.
I pass this question.
S: Let's come back to our previous topic.
Publication set 11 is the publication set.
It is involved in the research group.
Do you want a site presenting it?
U: No.
S: Do you want a page presenting it?
U: Yes.
S: A page presents the publication set.
Which page presents the publication set?
U: A new page.
S: Page 51 presents the publication set.

What is the title of page 51?
U: publication set.
S: Do you want a home page linked to it?
U: Yes.
S: A home page is linked to publication set page 51.
Which home page is linked to publication set page 51 ?
U: home page 24.
S: Do you want another page linked from publication set page 51?
U: No.
S: Do you want publication set page 51 linked from another home page ?
U: No.
S: Do you want a page linked to it?
U: No.
S: Do you want a home page linked from it?
U: No.
S: Do you want a navigator displayed by it?
U: Yes.
S: A navigator is displayed by publication set page 51.
Which navigator is displayed by publication set page 51 ?
U: A new navigator.
S: the navigator is displayed by publication set page 51.
Do you want the navigator displayed by another page ?
U: I'll answer this question later.
I pass this question.
S: Do you want a home page displaying it?
U: I'll answer this question later.
I pass this question.
S: Do you want a page pointed by it?
U: I'll answer this question later.
I pass this question.
S: Do you want a home page pointed by it?
U: I'll answer this question later.
I pass this question.
S: Do you want publication set page 51 pointed by the navigator ?
U: I'll answer this question later.
I pass this question.
S: Do you want research group home page 24 linked to another page ?
U: I'll answer this question later.
I pass this question.
S: Do you want a page linked to research group home page 24?
U: No.
S: Do you want a home page linked from it?
U: No.
S: Do you want a home page linked to it?
U: No.
S: Do you want the navigator pointing to it?
U: Yes.

S: The navigator is pointing to research group home page 24.
Do you want research group home page 24 displaying the navigator ?
U: No.
S: Let's come back to the questions that are still unanswered.
Do you want another page linked from it?
U: No.
S: Do you want the navigator pointing to a page ?
U: No.
S: Do you want the navigator displayed by another page ?
U: Yes.
S: A page displays the navigator.
Which page displays the navigator?
U: page 31.
S: Do you want the navigator pointing to another home page ?
U: No.
S: Do you want the navigator displayed by another page ?
U: Yes.
S: A page displays the navigator.
Which page displays the navigator ?
U: page 43.
S: Do you want a home page displaying the navigator?
U: No.
S: Do you want the navigator displayed by another page?
U: No.
S: Do you want project set page 43 pointed by the navigator?
U: No.
S: Do you want project set page 43 linked from another home page?
U: No.
S: Do you want researcher set page 31 linked to a home page?
U: I don't remember the identifier.
I pass this question.
S: Do you want researcher set page 31 pointed by the navigator?
U: No.
S: Do you want a page linked to it?
U: No.
S: Do you want publication set page 51 pointed by the navigator?
U: No.
S: Do you want researcher set page 31 linked to a home page ?
U: No.

Appendix C

Evaluation

C.1 Site structure

Requirements to give to the system :
The research group is presented by a site. This site is described by a home page. The researcher set, project set and publication set are presented by pages (one for each set). Each page and the home page have for title the concept they present. Each page displays a navigator. This navigator points to the research group home page. The research group home page has links to each other page.
Graphical representation of the requirements :
Research group → site → homepage
Researcher set → page + navigator
Project set → page + navigator
Publication set → page + navigator
Navigator → pointing to homepage
Homepage → linked to all three pages

C.2 Questionnaire

Please answer these questions based on your latest dialogue with the system. For specific comments on the dialogue, refer to the sentence numbers given by the replay_dialogue function.

- For how long have you been using this system before this dialogue?
  never / less than 5 min. / between 5 and 20 min. / more than 20 min.
- Was it easy to state the requirements?
  not at all / no / somewhat / yes / yes very much
- Was the dialogue related to the task you were performing?
  rarely / sometimes / usually / often / almost always

- Was the dialogue easy to understand?
  not at all / no / somewhat / yes / yes very much
- Was the order of the communications (presentations and questions) correct?
  rarely / sometimes / usually / often / almost always
- Additional Comments:

C.3 Evaluation grid

C.3.1 Filling the grids

Table C.1 Evaluation grid

	Dialogue 1	Dialogue 2	Dialogue 3	Dialogue 4	Dialogue 5	Dialogue 6
Dialogue1	1					
Dialogue2	X	1				
Dialogue3	X	X	1			
Dialogue4	X	X	X	1		
Dialogue5	X	X	X	X	1	
Dialogue6	X	X	X	X	X	1

Fill the cell at line i and row j with:

1 If Dialogue i and Dialogue j are equivalent (for the criterion under consideration)

3 If Dialogue i is somewhat better than Dialogue j

5 If Dialogue i is better than Dialogue j

7 If Dialogue i is much better than Dialogue j

9 If Dialogue i is outstanding compared to Dialogue j

1/3 If Dialogue j is somewhat better than Dialogue i

1/5 If Dialogue j is better than Dialogue i

1/7 If Dialogue j is much better than Dialogue i

1/9 If Dialogue j is outstanding compared to Dialogue i

C.3.2 Global coherence

Which dialogue is *globally* more coherent, i.e., is divided into well defined chunks that are correctly related together?
Globally coherent dialogues should focus on something for a while and then move on a related topic. Here are examples of global coherence and non-coherence.
Coherent:

Table C.2 Global coherence grid

	Dialogue 1	Dialogue 2	Dialogue 3	Dialogue 4	Dialogue 5	Dialogue 6
Dialogue1	1					
Dialogue2	X	1				
Dialogue3	X	X	1			
Dialogue4	X	X	X	1		
Dialogue5	X	X	X	X	1	
Dialogue6	X	X	X	X	X	1

> B: How do you teach your students to use a calculator?
> A: I think students should use a calculator for a while. I give them problems to solve with it, and when they have trouble, I answer their questions about the problems.
> B: That's all well and good, but I think they need more instruction on the device to reduce the number of questions. Instead I give them instructions, and they use these to solve problems. They don't have much trouble learning to use the machine.

Not Coherent:

> B: How do you teach your students to use a calculator?
> A: I think students should use a calculator for a while. I give them problems to solve with it, and when they have trouble, I answer their questions about the problems.
> B: Well, I think you are wrong. Here's why. I'm going on a vacation to Tahiti tomorrow. I'm going by plane, and I'll be there about a week. It is going to cost me a bundle of money.

More paticularly, if we compare the beginning of dialogue 1 wih the beginnings of dialogues 5 or 6 (see table C.3), we may find that the latter are less globally coherent because they move around too much. Other problems may be, for example, the fact of never jumping back to topics that should be dealt with or jumping without warning. The comparison value depends on your perception of the overall difference between the two dialogues on this aspect. The more you perceive the difference, the bigger the value should be.
(The answers of the user should not be considered. In particular, the fact that some communications are passed, i.e., not answered immediately after they are asked, should not be taken into account.)

C.3.3 Local coherence

Which dialogue is *locally* more coherent, i.e., has well related successive sentences? A dialogue is locally coherent if when sliding a window spanning two sentences over the text, the visible part is still related. Here are examples of local coherence and non-coherence.
Coherent:

> A: I just bought a new hat.
> B: It's nice. Where did you buy it?
> A: In the shop at the corner.

Table C.3 Global coherence example

Dialogue 1	Dialogue6
...	...
research group 1 is a research group	project set 15 is involved in research group 1
Do you want it presented by a page?	research group 1 is a research group
No	researcher set 7 is involved in it
Do you want it presented by a site?	researcher set 7 is a researcher set
Yes	Do you want it presented by a site?
research group 1 is presented by a site	No
Which site is presenting research group 1?	Do you want it presented by a page?
A new site	Yes
research group 1 is presented by site 22	researcher set 7 is presented by a page
Do you want site 22 described by a homepage?	Which page is presenting researcher set 7?
Yes	A new page
...	...

Table C.4 Local coherence grid

	Dialogue 1	Dialogue 2	Dialogue 3	Dialogue 4	Dialogue 5	Dialogue 6
Dialogue1	1					
Dialogue2	X	1				
Dialogue3	X	X	1			
Dialogue4	X	X	X	1		
Dialogue5	X	X	X	X	1	
Dialogue6	X	X	X	X	X	1

Not Coherent:

A: I just bought a new hat.
B: Fred eats hamburgers.
A: My car is fast.

More paticularly, if we compare dialogue 1 with dialogue 5 (see table C.5), we can see that questions in dialogue 5 shift from, for example, page to homepage whereas they do not in dialogue 1. Therefore, dialogue 1 is more locally coherent than dialogue 5 at that point. (The same may of course happen with other entities then page and homepage.)

Table C.5 Local coherence example

Dialogue 1	Dialogue5
...	...
Do you want it linked to a page?	Do you want project set page 38 linked from a page?
No	No
Do you want it linked from a page?	Do you want project set page 38 linked from a homepage?
No	Yes
Do you want it linked to a homepage?	Do you want project set page 38 linked to another page?
No	No
Do you want it linked from a homepage?	Do you want project set page 38 linked to a homepage?
Yes	No
...	...